

Event-Driven Receding Horizon Control For Distributed Persistent Monitoring on Graphs

Shirantha Welikala and Christos G. Cassandras

Abstract—We consider the optimal multi-agent persistent monitoring problem defined on a set of nodes (targets) interconnected through a fixed graph topology. The objective is to minimize a measure of mean overall node state uncertainty evaluated over a finite time interval by controlling the motion of a team of agents. Prior work has addressed this problem through on-line parametric controllers and gradient-based methods, often leading to low-performing local optima or through off-line computationally intensive centralized approaches. This paper proposes a computationally efficient event-driven receding horizon control approach providing a distributed on-line gradient-free solution to the persistent monitoring problem. A novel element in the controller, which also makes it parameter-free, is that it self-optimizes the planning horizon over which control actions are sequentially taken in event-driven fashion. Numerical results show significant improvements compared to state of the art distributed on-line parametric control solutions.

I. INTRODUCTION

A *persistent monitoring* problem arises when a dynamically changing environment is monitored by a set of mobile agents and encompasses applications such as environmental sensing [1], surveillance systems [2], energy management [3] and data collection [4]. In contrast to cases where every point in the environment is equally valued for agents to monitor, in many others, only a finite set of “points of interest” (henceforth called “targets”) holds a positive value [5], [6]. The persistent monitoring problem considered in this paper belongs to the latter class, where the goal of the agent team is to monitor (sense or collect information from) each target in order to reduce an “uncertainty metric” associated with the target state. Typically, this uncertainty metric increases when no agent is monitoring the target and decreases when one or more agents are able to monitor it by dwelling in its vicinity. The global objective is to control the agent movement so as to minimize an overall measure of target uncertainties.

Persistent monitoring problems in 1D environments have been solved using classical optimal control techniques. For such problems, the optimal solutions have been shown to be threshold-based parametric controllers [7]. However, this synergy between optimal control and parametric controllers does not extend to 2D environments [8]. Nevertheless, one can still optimize agent trajectories within parametric families [8], [9] (e.g., elliptical). Apart from the apparent sub-optimality, failing to react to dynamic changes in target uncertainties and the dependence of performance on the

initial target/agent conditions are drawbacks of this approach. As a solution, recent work [5] has proposed a graph abstraction (where targets are modeled as nodes and inter-target agent trajectory segments are modeled as edges) to formulate *Persistent Monitoring on Graphs* (PMG) problems.

In PMG problems, an agent trajectory is defined by the sequence of *targets to be visited* and the *dwell time* spent at each visited target. To overcome the complexity of this problem, a *distributed* Threshold-based Control Policy (TCP) is adopted in [5], where each agent enforces a set of thresholds on its neighboring target uncertainty values to make immediate trajectory decisions: the dwell time to be spent and the next target to visit. The threshold values are then optimized using an *on-line* gradient-based technique based on Infinitesimal Perturbation Analysis (IPA) [10]. However, this IPA-TCP approach often converges to poor locally optimal solutions. As a remedy, [6] has proposed to append an *off-line centralized* threshold initialization scheme which is shown to considerably increase performance at the expense of significant computational effort.

Motivated by these challenges, this paper presents an entirely different approach that can be used to solve the same PMG problem. Specifically, the event-driven nature of PMG systems is exploited to derive an Event-Driven Receding Horizon Controller (ED-RHC) to optimally govern each of the agents in an on-line distributed manner using only a minimal amount of computational power. First, it is shown that each agent’s trajectory is fully characterized by the sequence of decisions it makes at specific discrete event times. Second, for any agent, a *Receding Horizon Control Problem* (RHCP) is formulated to determine locally optimal decisions over a planning horizon, to be executed only over a shorter action horizon, with the process sequentially repeated as new events take place. In contrast to the prior ED-RHC approaches [11], [9], [12] and gradient-based approaches [5], [6], the proposed ED-RHC in this paper is both parameter-free and gradient-free as shown in the sequel.

II. PROBLEM FORMULATION

Consider an n -dimensional mission space containing M targets (nodes) in the set $\mathcal{T} = \{1, 2, \dots, M\}$ where the location of target i is fixed at $Y_i \in \mathbb{R}^n$. A team of N agents in the set $\mathcal{A} = \{1, 2, \dots, N\}$ is deployed to monitor the targets. Each agent $a \in \mathcal{A}$ moves within this space and its location at time t is denoted by $s_a(t) \in \mathbb{R}^n$.

a) **Target Model:** Each target $i \in \mathcal{T}$ has an associated *uncertainty state* $R_i(t) \in \mathbb{R}$ which follows the dynamics:

*Supported in part by NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, by AFOSR under grant FA9550-19-1-0158, by ARPA-E’s NEXTCAR program under grant DE-AR0000796 and by the MathWorks.

The authors are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, {shiran27, cgc}@bu.edu.

$$\dot{R}_i(t) = \begin{cases} A_i - B_i N_i(t) & \text{if } R_i(t) > 0 \text{ or } A_i - B_i N_i(t) > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $N_i(t) = \sum_{a \in \mathcal{A}} \mathbf{1}\{s_a(t) = Y_i\}(\mathbf{1}\{\cdot\})$ is the indicator function) and the values of $A_i, B_i, R_i(0)$ are prespecified. Therefore, $N_i(t)$ is the number of agents present at target i at time t . According to (1): (i) $R_i(t)$ increases at a rate A_i when no agent is visiting target i , (ii) $R_i(t)$ decreases at a rate $B_i N_i(t) - A_i$ where B_i is the uncertainty removal rate by a visiting agent (i.e., agent sensing or data collection rate) to the target i , and, (iii) $R_i(t) \geq 0, \forall t$. This problem set-up has an attractive queueing system interpretation [5] where A_i and $B_i N_i(t)$ are respectively thought of as the arrival rate and the controllable service rate at target (server) $i \in \mathcal{T}$ in a queueing network.

b) Agent Model: Some persistent monitoring models (e.g., [7], [13]) assume each agent $a \in \mathcal{A}$ to have a finite sensing range $r_a > 0$ allowing it to decrease $R_i(t)$ whenever it is in the vicinity of target $i \in \mathcal{T}$ (i.e., when $\|s_a(t) - Y_i\| \leq r_a$). Since we will adopt a graph topology for this problem, the condition $\|s_a(t) - Y_i\| \leq r_a$ is represented by the agent residing at the i th vertex of a graph and $N_i(t)$ is used to replace the role of the joint detection probability of a target i used in [7], [13]. Moreover, similar to [6] the analysis in this paper is independent of the agent motion dynamic model.

c) Objective: Our objective is to minimize the *mean system uncertainty* J_T over a finite time interval $t \in [0, T]$:

$$J_T \triangleq \frac{1}{T} \int_0^T \sum_{i \in \mathcal{T}} R_i(t) dt, \quad (2)$$

by controlling the motion of the agents through a suitable set of feasible controls to be described in the sequel.

d) Graph Topology: A directed graph topology $\mathcal{G} = (\mathcal{T}, \mathcal{E})$ is embedded into the mission space such that the *targets* are represented by the graph *vertices* $\mathcal{T} = \{1, 2, \dots, M\}$, and the inter-target *trajectory segments* are represented by the graph *edges* $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{T}\}$. We point out that these trajectory segments in $\mathbb{R}^n$ may take arbitrary shapes so as to account for potential constraints in the agent motion; in the graph $\mathcal{G}$, each segment represented by an edge $(i, j) \in \mathcal{E}$ has an associated value $\rho_{ij} \in \mathbb{R}_{\geq 0}$ representing the *transit time* an agent spends to travel from target i to j . The *neighbor set* and the *neighborhood* of a target $i \in \mathcal{T}$ are defined respectively as $\mathcal{N}_i \triangleq \{j : (i, j) \in \mathcal{E}\}$ and $\tilde{\mathcal{N}}_i = \mathcal{N}_i \cup \{i\}$.

e) Control: Based on this embedded graph topology $\mathcal{G}$, whenever an agent $a \in \mathcal{A}$ is ready to leave a target $i \in \mathcal{T}$, it selects a *next-visit* target $j \in \mathcal{N}_i$. Hence, the agent travels over $(i, j) \in \mathcal{E}$ to arrive at target j after an amount of time ρ_{ij} . Subsequently, it selects a *dwelt-time* $\tau_j \in \mathbb{R}_{\geq 0}$ to spend at target j (which contributes to decreasing $R_j(t)$), and then makes another next-visit decision.

Therefore, in a PMG problem the control exerted consists of a sequence of *next-visit* targets $j \in \mathcal{N}_i$ and *dwelt-times* $\tau_i \in \mathbb{R}_{\geq 0}$. Our goal is to determine (τ_i, j) for any agent residing

at i at any time $t \in [0, T]$ which are optimal in the sense of minimizing (2). As pointed out in [5], this is a challenging task even for the simplest PMG problem configurations due to the nature of the search space.

f) Receding Horizon Control: The on-line distributed IPA-TCP method proposed in [5] requires each agent to use a set of *thresholds* applied to its neighborhood target uncertainties $\{R_j(t) : j \in \mathcal{N}_i\}$ in order to determine its dwell-time and next-visit decisions. Thus, the objective in (2) is viewed as dependent on these threshold parameters. Starting from an arbitrary set of thresholds, each agent iteratively adjusts them using a gradient technique that exploits the information from observed events in agents' trajectories. Although this approach is efficient due to the use of IPA, it is limited by the presence of local optima.

To address this limitation, this paper proposes an *Event-Driven Receding Horizon Controller* (ED-RHC) at each agent $a \in \mathcal{A}$. The basic idea of RHC has its root in Model Predictive Control (MPC) but, in addition, it exploits the event-driven nature of the PMG problem to reduce complexity by orders of magnitude, provide flexibility in the frequency of control updates, and improve performance by avoiding many local optima resulting from gradient-based optimization. As introduced in [11] and extended in [9],[12], ED-RHC solves an optimization problem of the form (2) but limited to a given *planning horizon* whenever an event is observed; the resulting control is then executed over a generally shorter *action horizon* defined by the occurrence of the next event of interest to the controller. This process is iteratively repeated in event-driven fashion. In the PMG problem, the aim of the ED-RHC when invoked at time t with an agent residing at $i \in \mathcal{T}$ is to determine the immediate next-visit $j \in \mathcal{N}_i$ and dwell times at i, j , jointly forming a control $U_i(t)$. This is done by solving an optimization problem of the form:

$$U_i^*(t) = \arg \min_{U_i(t) \in \mathbb{U}(t)} [J_H(X_i(t), U_i(t); H) + \hat{J}_H(X_i(t+H))], \quad (3)$$

where $X_i(t)$ is the current local state and $\mathbb{U}(t)$ is the feasible control set at t . The term $J_H(X_i(t), U_i(t); H)$ is the immediate cost over the planning horizon $[t, t+H]$ and $\hat{J}_H(X_i(t+H))$ is an estimate of the future cost evaluated at the end of the planning horizon $t+H$. The value of H is selected in prior work [11],[9],[12] *exogenously*. However, in this paper we will include this value into the optimization problem and ignore the $\hat{J}_H(X_i(t+H))$ term. Thus, by optimizing the planning horizon we compensate for the complexity and intrinsic inaccuracy of $\hat{J}_H(X_i(t+H))$. Further, this modification enables solving (3) in closed form. Thus, the proposed ED-RHC is both parameter-free and gradient-free. Moreover, the proposed ED-RHC is also *distributed* as it allows each agent to separately solve (3) using only local state information.

A. Preliminary Results

According to (1), the target state $R_i(t), i \in \mathcal{T}$, is piece-wise linear and its gradient $\dot{R}_i(t)$ changes only when one of the following (strictly local) *events* occurs: (i) An agent arrival

at i , (ii) $[R_i(t) \rightarrow 0^+]$, or (iii) An agent departure from i . Let the occurrence of such events be indexed by $k = 1, 2, \dots$ with an associated occurrence time t_i^k with $t_i^0 = 0$. Then,

$$\dot{R}_i(t) = \dot{R}_i(t_i^k), \quad \forall t \in [t_i^k, t_i^{k+1}). \quad (4)$$

Remark 1: Allowing agents to have overlapping dwell sessions at some target (also known as “target sharing”) is widely regarded as inefficient [14], [5], [6]. This motivates us to enforce a constraint on the controller to ensure:

$$N_i(t) \in \{0, 1\}, \quad \forall t \in [0, T], \quad i \in \mathcal{T}. \quad (5)$$

Clearly, this constraint only applies if $N \geq 2$.

Under the constraint (5), it follows from (1) and (4) that the sequence $\{\dot{R}_i(t_i^k)\}_{k=0,1,\dots}$ is a *cyclic order* of three elements: $\{-(B_i - A_i), 0, A_i\}$. Next, to make sure that each agent is capable of enforcing the event $[R_i \rightarrow 0^+]$ at any $i \in \mathcal{T}$, the following simple stability condition [6] is assumed.

Assumption 1: Target uncertainty rate parameters A_i and B_i of each target $i \in \mathcal{T}$ satisfy $0 < A_i < B_i$.

a) Decomposition of the objective function: The following Theorem 1 provides a target-wise and temporal decomposition of the main objective function J_T in (2) (due to space limitations, all proofs are provided in [15]).

Theorem 1: The contribution to the main objective J_T by a target $i \in \mathcal{T}$ during a time period $[t_0, t_1) \subseteq [t_i^k, t_i^{k+1})$ for some $k \in \mathbb{Z}_{\geq 0}$ is $\frac{1}{T} J_i(t_0, t_1)$, where,

$$J_i(t_0, t_1) = \int_{t_0}^{t_1} R_i(t) dt = \frac{(t_1 - t_0)}{2} [2R_i(t_0) + \dot{R}_i(t_0)(t_1 - t_0)].$$

A simple corollary of Theorem 1 is to extend it to any interval $[t_0, t_1)$ which may include one or more event times t_i^k (the proof is straightforward and found in [15]).

b) Local objective function: The *local objective function* of target i over a period $[t_0, t_1) \subseteq [0, T]$ is defined as

$$\bar{J}_i(t_0, t_1) = \sum_{j \in \mathcal{N}_i} J_j(t_0, t_1). \quad (6)$$

The value of each $J_j(t_0, t_1)$ above is obtained through Theorem 1 and its extension (see [15]) if $[t_0, t_1)$ includes additional events: $[t_0, t_1)$ is decomposed into a sequence of corresponding inter-event time intervals.

B. ED-RHC optimization problem formulation

Let agent $a \in \mathcal{A}$ reside at target $i \in \mathcal{T}$ at some $t \in [0, T]$. In our distributed setting, we assume that agent a is made aware of only *local events* occurring in the neighborhood $\mathcal{N}_i$. As mentioned earlier, the control $U_i(t)$ consists of the *dwell time* τ_i at the current target i , and the *next target* $j \in \mathcal{N}_i$ to visit. Once j is known, then the agent can also determine the *dwell time* τ_j at the next target j . Moreover, a dwell time decision τ_i (or τ_j) can be divided into two interdependent decisions: (i) the *active time* u_i (or u_j) and (ii) the *inactive* (or *idle*) time v_i (or v_j), as shown in Fig. 1. Thus, agent a has to optimally choose five *decision variables* which form the control vector $U_i(t) = [u_i(t), v_i(t), j, u_j(t), v_j(t)]$.

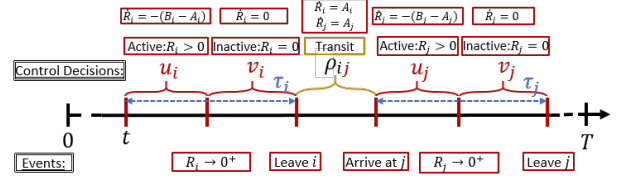


Fig. 1: Event timeline and control decisions under RHC.

a) Planning Horizon: Recalling (3), the ED-RHC depends on the planning horizon $H \in \mathbb{R}_{\geq 0}$ which is viewed as a *fixed* control parameter. Note that $t + H$ is constrained by $t + H \leq T$, hence if this is violated we redefine the planning horizon to be $H = T - t$. For simplicity, in what follows we omit this situation which only arises as the process approaches the terminal time T .

Let us decompose the control $U_i(t)$ into its real-valued components and its discrete (target index $j \in \mathcal{N}_i$) component. Thus (omitting time arguments), set $U_{ij} = [u_i, v_i, u_j, v_j]$ and let the current local state be $X_i(t) = \{R_j(t) : j \in \mathcal{N}_i\}$. Then, the optimal controls are obtained by solving the following set of optimization problems, henceforth called the ED-RHC Problem (RHCP):

$$U_{ij}^* = \arg \min_{U_{ij} \in \mathbb{U}} J_H(X_i(t), U_{ij}; H); \quad \forall j \in \mathcal{N}_i, \text{ and}, \quad (7)$$

$$j^* = \arg \min_{j \in \mathcal{N}_i} J_H(X_i(t), U_{ij}^*; H). \quad (8)$$

Note that (7) involves solving a number $|\mathcal{N}_i|$ of optimization problems, one for each $j \in \mathcal{N}_i$. Then, (8) determines j^* through a simple numerical comparison. Therefore, the final optimal decision variables are $U_{ij^*}^*$ and j^* . Here, the notation $|\cdot|$ denotes the 1-norm or the cardinality operator when the argument is respectively a vector or a set.

The objective function $J_H(\cdot)$ above is chosen to reflect the contribution to the main objective J_T in (2) by the targets in the neighborhood $\mathcal{N}_i$ over the time period $[t, t + H]$, which is provided by (6) and Theorem 1 as

$$J_H(X_i(t), U_{ij}; H) = \frac{1}{H} \bar{J}_i(t, t + H). \quad (9)$$

The *feasible control space* $\mathbb{U}$ in (7) is such that u_i, v_i, u_j, v_j are non-negative real variables. In addition, note that if $|U_{ij}| + \rho_{ij} = H$ (see also Fig.1) then this restricts the choices of the four control variables. Thus, the selection of H affects the problem's optimal solution. For example, if H is very large (or very small), clearly the optimal decisions $U_{ij^*}^*$ and j^* are not globally optimal. Attempting to find the optimal choice of H without compromising the on-line distributed nature of the ED-RHC solution is a challenging task.

To address this problem, we introduce a *variable horizon* w defined as:

$$w \triangleq |U_{ij}| + \rho_{ij} = u_i + v_i + \rho_{ij} + u_j + v_j, \quad (10)$$

and replace H in (9) by w while, at the same time, imposing the constraint: $w \leq H$. Therefore, the objective function J_H and the feasible control space $\mathbb{U}$ in the RHCP are chosen as

$$J_H(X_i(t), U_{ij}; H) = \frac{1}{w} \bar{J}_i(t, t + w), \text{ and}, \quad (11)$$

$$\mathbb{U} = \{U : U \in \mathbb{R}^4, U \geq 0, |U| + \rho_{ij} \leq H\}.$$

Note that, the constraint $|U_{ij}| + \rho_{ij} \leq H$ above ensures $w \leq H$. Moreover, this novel RHCP formulation allows us to simultaneously determine the *optimal planning horizon* size $w^* = |U_{ij}^*| + \rho_{ij}^*$ in terms of the optimal control $U_i^*(t)$.

On the other hand, having a control-dependent denominator term in the objective function and adding an extra dimension to the feasible control space of the RHCP introduce new technical challenges that we address in the rest of the paper. To accomplish this, we will exploit structural properties of (11) and show that the RHCP in (7) can be solved analytically and efficiently.

b) Event-Driven Action Horizon: As in all RHCPs, the solution of each optimization problem over a planning horizon H is executed only over an *action horizon* $h \leq H$. In the ED-RHC setting, the value of h is determined by the first event that takes place after t (when the RHCP was last solved). Thus, in contrast to time-driven RHC, the control is updated whenever asynchronous events occur; this prevents unnecessary steps to re-solve the RHCP (7)-(8) with (11).

In general, the determination of the action horizon h may be controllable or uncontrolled. The latter case occurs as a result of random events in the system (if such events are part of the setting), while the former corresponds to the occurrence of any one event whose occurrence results from an agent solving a RHCP. We define next the three *controllable* events associated with an agent when it resides at target i ; each of these events defines the action horizon h following the solution of a RHCP by this agent at some time $t \in [0, T]$:

1. Event $[h \rightarrow u_i^*]$: This event occurs at time $t + u_i^*(t)$. If $R_i(t + u_i^*(t)) = 0$, it coincides with an $[R_i \rightarrow 0^+]$ event. Otherwise, i.e., if $R_i(t + u_i^*(t)) > 0$, this implies that the solution of the associated RHCP is to terminate the active time at target i before the $[R_i \rightarrow 0^+]$ event. Therefore, by definition, no inactive time may follow, i.e., $v_i^*(t) = 0$, and $[h \rightarrow u_i^*]$ coincides with a departure event from target i .

2. Event $[h \rightarrow v_i^*]$: This event occurs at time $t + v_i^*(t)$. It is only feasible after an event $[h \rightarrow u_i^*]$ has occurred, including the possibility that $u_i^*(t) = 0$ in the RHCP solution determined at t . Clearly, this coincides with a departure event from target i .

3. Event $[h \rightarrow \rho_{ij}^*]$: This event occurs at time $t + \rho_{ij}^*$. It is only feasible after an event $[h \rightarrow u_i^*]$ or $[h \rightarrow v_i^*]$ has occurred, including the possibility that $u_i^*(t) = 0$ and $v_i^*(t) = 0$ in the RHCP solution determined at t . Clearly, this coincides with an arrival event at target j^* as determined by the RHCP solution obtained at t .

Observe that these events are mutually exclusive, i.e., only one is feasible at any one time. It is also possible for a different event to occur after t and before one of these occurs; such an event is either random (if our model allows for such events) or it is controllable but associated with a different target than i . In particular, let us define two additional events that may occur at any neighbor $j \in \mathcal{N}_i$ and affect the agent residing at i . These events are a consequence of control constraint (5) and pertain only to multi-agent persistent monitoring problems, where our controller must

enforce the no-target-sharing policy.

A target $j \in \mathcal{T}$ is said to be *covered* at time t if it already has a residing agent or if an agent is en route to visit it from a neighboring target in $\mathcal{N}_j$. Thus, j is covered only if $\exists k \in \mathcal{N}_j$ and $\tau \in [t, t + \rho_{kj})$ such that $\sum_{a \in \mathcal{A}} \mathbf{1}\{s_a(\tau) = Y_j\} > 0$. Since neighboring targets communicate with each other, this information can be determined at any target in $\mathcal{N}_j$ at any time t . Therefore, an agent $a \in \mathcal{A}$ residing at target i can prevent target sharing at $j \in \mathcal{N}_i$ by simply modifying the neighbor set $\mathcal{N}_i$ used in the RHCP solved at t to exclude all covered targets. Let us use $\mathcal{N}_i(t)$ to indicate a *time-varying* neighborhood of i . Then, if target j becomes covered at t , we set $\mathcal{N}_i(t) = \mathcal{N}_i(t^-) \setminus \{j\}$. The effect of this modification is clear if a RHCP solved by an agent at target i at some time t leads to a next visit solution $j^* \in \mathcal{N}_i$, and if this is followed by an event at $t' > t$ causing j^* to be covered, then $\mathcal{N}_i(t') = \mathcal{N}_i(t) \setminus \{j^*\}$ and the agent at i (whether active or inactive) must re-solve the RHCP at t' with the new $\mathcal{N}_i(t')$. Note that as soon as an agent a is en route to j^* , then j^* becomes covered, hence preventing any other agent from visiting j^* prior to agent a 's subsequent departure from j^* .

Based on this discussion, we define the following two additional *neighbor-induced local events* at $j \in \mathcal{N}_i$ affecting an agent a residing at target i :

4. Covering Event C_j , $j \in \mathcal{N}_i$: This event causes $\mathcal{N}_i(t)$ to be modified to $\mathcal{N}_i(t) \setminus \{j\}$.

5. Uncovering Event $\bar{C}_j$, $j \in \mathcal{N}_i$: This event causes $\mathcal{N}_i(t)$ to be modified to $\mathcal{N}_i(t) \cup \{j\}$.

If one of these two events takes place while an agent residing at target i is either active or inactive, then the RHCP (7)-(8) is re-solved to account for the updated $\mathcal{N}_i(t)$. This may affect the values of the optimal solution U_i^* from the previous solution. Note, however, that the new solution will still give rise to an event $[h \rightarrow u_i^*]$ (if the RHCP is solved while the agent is active) or $[h \rightarrow v_i^*]$ (if the RHCP is solved while the agent is inactive).

It is clear from this discussion that the exact form of the RHCP to be solved at time t depends on the event that triggers the end of an action horizon. In particular, there are three possible forms of the RHCP (7)-(8):

RHCP1: This problem is solved by an agent when an event $[h \rightarrow \rho_{ki}]$ occurs at time t for any $k \in \mathcal{N}_i(t)$, i.e., the agent arrives at target i . The solution $U_i^*(t)$ includes $u_i^*(t) \geq 0$, representing the amount of time that the agent should be active at i . This problem may also be solved while the agent is active at i if a C_j or $\bar{C}_j$ event occurs for any $j \in \mathcal{N}_i(t)$.

RHCP2: This problem is solved by an agent residing at i when an event $[h \rightarrow u_i^*]$ occurs at time t and $R_i(t) = 0$. The solution $U_i^*(t)$ is now constrained to include $u_i^*(t) = 0$ by default, since the agent can no longer be active at i . This problem may also be solved while the agent is inactive at i if a C_j or $\bar{C}_j$ event occurs for any $j \in \mathcal{N}_i(t)$.

RHCP3: This problem is solved by an agent residing at i when an event $[h \rightarrow u_i^*]$ occurs at time t and $R_i(t) > 0$. The solution $U_i^*(t)$ is again constrained to include $u_i^*(t) = 0$ by default; in addition, it is constrained to have $v_i^*(t) = 0$ since the agent ceases being active while $R_i(t) > 0$, implying that it

must immediately depart from i without becoming inactive. This problem is also solved when an event $[h \rightarrow v_i^*]$ occurs at time t , implying that the agent departs from i .

III. SOLVING THE EVENT-DRIVEN RECEDING HORIZON CONTROL PROBLEMS

We begin with **RHCP3** because it is the simplest problem given that in this case $u_i^*(t) = 0$ and $v_i^*(t) = 0$ by default and U_{ij} in (7) is limited to $U_{ij} = [u_j, v_j]$.

1) *Solution of RHCP3*: In this case, the variable horizon w in (10) becomes $w = \rho_{ij} + u_j + v_j$ and $\rho_{ij} \leq w \leq H$ where H is the fixed planning horizon. Therefore, any targets $j \in \mathcal{N}_i(t)$ such that $\rho_{ij} > H$ are omitted from (7).

a) *Constraints*: We begin by identifying a tight upper bound for the active time control variable $u_j(t)$. This is defined by the maximum active time possible to spend on target j , which is given by $R_j(t + \rho_{ij} + u_j) = 0$. Denoting this bound by u_j^B , it follows from (1) that $u_j^B = \frac{R_j(t + \rho_{ij})}{B_j - A_j} = \frac{R_j(t) + A_j \rho_{ij}}{B_j - A_j}$. Note that even though we should write $u_j^B(t)$ since it depends on the initial target uncertainty $R_j(t)$, we omit this time dependence for notational simplicity. Moreover, note that in order to have a positive inactive time $v_j > 0$, a necessary condition is that it first spends the maximum active time possible $u_j = u_j^B$. We now see that any feasible pair (u_j, v_j) in (7) belongs to one of the two constraint sets: $\mathbb{U}_1 = \{0 \leq u_j \leq u_j^B, v_j = 0\}$ and $\mathbb{U}_2 = \{u_j = u_j^B, v_j \geq 0\}$, where $(u_j^B, 0)$ is allowed to be a feasible control in both sets. An additional constraint is imposed by the variable horizon $w = \rho_{ij} + u_j + v_j \leq H$. Thus, we define $\bar{u}_j \triangleq \min\{u_j^B, H - \rho_{ij}\}$ and $\bar{v}_j \triangleq H - (\rho_{ij} + u_j^B)$ where, $\bar{u}_j$ and $\bar{v}_j$ are the maximum possible values of u_j and v_j respectively. Then, the two constraint sets become

$$\mathbb{U}_1 = \{0 \leq u_j \leq \bar{u}_j, v_j = 0\} \text{ and } \mathbb{U}_2 = \{u_j = u_j^B, 0 \leq v_j \leq \bar{v}_j\}. \quad (12)$$

Therefore, (12) is the feasible control set for $U_{ij} = [u_j, v_j]$.

b) *Objective*: Following (11), the objective function for **RHCP3** is $J_H(U_{ij}) = J_H(X_i(t), [0, 0, U_{ij}]; H) = \frac{1}{w} \bar{J}_i(t, t + w)$. To obtain an exact expression for J_H , first the local objective function $\bar{J}_i$ is decomposed using (6):

$$\bar{J}_i = J_j + \sum_{m \in \mathcal{N}_i(t) \setminus \{j\}} J_m. \quad (13)$$

Both J_j and J_m in (13) are evaluated for the period $[t, t + w]$ using Theorem 1 and we get :

- $J_j = \frac{\rho_{ij}}{2} [2R_j(t) + A_j \rho_{ij}] + \frac{u_j}{2} [2(R_j(t) + A_j \rho_{ij}) - (B_j - A_j)u_j],$
- $J_m = \frac{(\rho_{ij} + u_j + v_j)}{2} [2R_m(t) + A_m(\rho_{ij} + u_j + v_j)].$

Combining the above two results with (13) gives the complete objective function $J_H(U_{ij})$ as

$$J_H(u_j, v_j) = \frac{C_1 u_j^2 + C_2 v_j^2 + C_3 u_j v_j + C_4 u_j + C_5 v_j + C_6}{\rho_{ij} + u_j + v_j},$$

where, $C_1 = \frac{1}{2}[\bar{A} - B_j]$, $C_2 = \frac{\bar{A}_j}{2}$, $C_3 = \bar{A}_j$, $C_4 = [\bar{R}(t) + \bar{A} \rho_{ij}]$, $C_5 = [\bar{R}_j(t) + \bar{A}_j \rho_{ij}]$, $C_6 = \frac{\rho_{ij}}{2} [2\bar{R}(t) + \bar{A} \rho_{ij}]$, and $\bar{A} = \sum_{m \in \mathcal{N}_i} A_m$, $\bar{R}(t) = \sum_{m \in \mathcal{N}_i} R_m(t)$, $\bar{A}_j = \bar{A} - A_j$, $\bar{R}_j = \bar{R} - R_i$. Note that all these coefficients are non-negative except for C_1 which is non-negative only when: $B_j \leq \bar{A}$.

c) *Solving RHCP3 for optimal control* (u_j^*, v_j^*): The solution (u_j^*, v_j^*) of (7) is given by

$$(u_j^*, v_j^*) = \arg \min_{(u_j, v_j)} J_H(u_j, v_j), \quad (14)$$

where $(u_j, v_j) \in \mathbb{U}_1$ or $(u_j, v_j) \in \mathbb{U}_2$ as in (12).

- **Case 1**: $(u_j, v_j) \in \mathbb{U}_1 = \{0 \leq u_j \leq \bar{u}_j, v_j = 0\}$: In this case, $v_j^* = 0$ and (14) takes the form:

$$u_j^* = \arg \min_{0 \leq u_j \leq \bar{u}_j} J_H(u_j, 0). \quad (15)$$

Lemma 1: The optimal solution for (15) is

$$u_j^* = \begin{cases} \bar{u}_j & \text{if } \bar{u}_j \geq u_j^\# \text{ and } \bar{A} < B_j \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

$$u_j^\# = \frac{\bar{A} \rho_{ij}}{B_j - \bar{A}}. \quad (17)$$

Note that $u_j^\#$ in (17) can be thought of as a break-even point for u_j , where when $\bar{u}_j$ allows u_j to go beyond such a $u_j^\#$ value, it is always optimal to do so by choosing the extreme point $u_j = \bar{u}_j$ as u_j^* .

- **Case 2**: $(u_j, v_j) \in \mathbb{U}_2 = \{u_j = u_j^B, 0 \leq v_j \leq \bar{v}_j\}$: In this case, $u_j^* = u_j^B$ and (14) takes the form:

$$v_j^* = \arg \min_{0 \leq v_j \leq \bar{v}_j} J_H(u_j^B, v_j). \quad (18)$$

Lemma 2: The optimal solution for (18) is

$$v_j^* = \begin{cases} 0 & \text{if } \bar{A} \geq B_j \left[1 - \frac{\rho_{ij}^2}{(\rho_{ij} + u_j^B)^2} \right] \\ \min\{v_j^\#, \bar{v}_j\} & \text{otherwise,} \end{cases} \quad (19)$$

$$v_j^\# = \sqrt{\frac{(B_j - A_j)(\rho_{ij} + u_j^B)^2 - B_j \rho_{ij}^2}{\bar{A}_j}} - (\rho_{ij} + u_j^B). \quad (20)$$

Note that, unlike $u_j^\#$ given in (17) for problem (15), $v_j^\#$ given in (20) for problem (18) is an optimal choice for v_j .

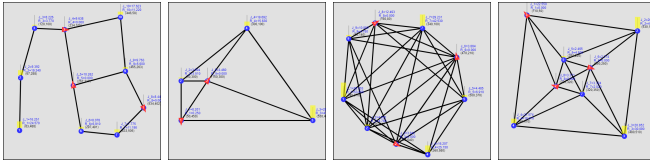
Theorem 2: The optimal solution of (14) is

$$(u_j^*, v_j^*) = \begin{cases} (0, 0) & \text{if } u_j^\# > \bar{u}_j \text{ or } \bar{A} \geq B_j \\ (\bar{u}_j, 0) & \text{else if } \bar{u}_j < u_j^B \\ (u_j^B, 0) & \text{else if } \bar{A} \geq B_j \left[1 - \frac{\rho_{ij}^2}{(\rho_{ij} + u_j^B)^2} \right] \\ (u_j^B, v_j^\#) & \text{else if } v_j^\# \leq \bar{v}_j \\ (u_j^B, \bar{v}_j) & \text{otherwise.} \end{cases} \quad (21)$$

Remark 2: The above theorem implies that whenever: (i) the planning horizon H is sufficiently large, (ii) the sensing capabilities are higher $B_j > \bar{A}$ and (iii) target uncertainty $R_j(t)$ is larger than some known threshold, it is optimal to plan ahead to drive the uncertainty level at target j to zero, $u_j^* = u_j^B$. This conclusion is in line with Theorem 1 in [5].

d) *Solving for optimal next target j^** : Using Theorem 2, when agent a is ready to leave target i , it can evaluate the optimal trajectory costs $J_H(u_j^*, v_j^*)$ for all $j \in \mathcal{N}_i(t)$. Based on the second step of the RHCP solved at t , i.e., (8), the optimal neighbor to visit next is $j^* = \arg \min_{j \in \mathcal{N}_i(t)} J_H(u_j^*, v_j^*)$. Thus, upon

solving **RHCP3**, agent a departs from target i at time t and follows the path $(i, j^*) \in \mathcal{E}$ to visit target j^* . In the spirit of



(a) $J_T = 105.5$. (b) $J_T = 63.7$. (c) $J_T = 114.1$. (d) $J_T = 97.2$.
(Impr.:+61.0%) (Impr.:+30.5%) (Impr.:+58.3%) (Impr.:+51.7%)

Fig. 2: Simulation examples with ED-RHC solution (state shown at terminal time $t = T$) and the percentage improvement with respect to the IPA-TCP solution.

RHC, recall that the optimal control will be updated upon the occurrence of the next event, which, in this case, will be the arrival of the agent at j^* , triggering the solution of an instance of **RHCPI**.

2) *Solution of RHCPI and RHCP2*: Due to space limitations, this topic is omitted here but can be found in [15].

3) *Complexity of RHCPs*: As shown in this work and [15], all three problem forms can be solved in closed form. Therefore, their complexity is constant and the overall RHC complexity scales linearly with the number of events in $[0, T]$.

IV. SIMULATION RESULTS

This section compares the performance J_T in (2) obtained for several different persistent monitoring problem configurations using: (i) the proposed ED-RHC method and (ii) the threshold control policy method (IPA-TCP) proposed in [5]. Note that both ED-RHC and IPA-TCP solutions are on-line and distributed (in contrast to [6]). All these solutions have been implemented in a JavaScript based simulator available at <http://www.bu.edu/codes/simulations/shiran27/PersistentMonitoring/>.

A compilation of extensive simulation results observed under a diverse set of scenarios is provided in [15]. Here, we limit ourselves to the four multi-agent PMG problem configurations shown in Fig. 2 (a)-(d). In each case, blue circles represent the targets, while black lines represent available path segments that agents can take to travel between targets. Red triangles and the yellow vertical bars indicate the agent locations and the target uncertainty levels, respectively. Moreover, since both of those quantities are time-varying (i.e., $s_a(t)$ and $R_i(t)$), in the figures only their state at the terminal time $t = T$ is shown when using the ED-RHC solution. In each problem configuration, the problem parameters are: $A_i = 1$, $B_i = 10$, $R_i(0) = 0.5$, $\forall i \in \mathcal{T}$ and target location co-ordinates (i.e., Y_i) are specified in each problem configuration figure. In all examples, targets have been placed inside a 600×600 mission space. The time period was taken as $T = 500$. Each agent's maximum speed was taken as 50 units/sec (under the first-order model [5]). The initial locations of the agents were chosen such that $s_a(0) = Y_i$ with $i = 1 + (a - 1) * \text{round}(M/N)$. The fixed planning horizon H was chosen as $H = 250$.

Each sub-figure caption in Fig. 2 provides the cost value J_T in (2) observed under the proposed ED-RHC approach and the percentage improvement achieved compared to the IPA-TCP method [5]. Based on these results (for more, see

[15]), the proposed ED-RHC method performs considerably better (on average 50.4% better) than the IPA-TCP method.

V. CONCLUSION

This paper considers the optimal multi-agent persistent monitoring problem defined on a set of targets interconnected according to a fixed graph topology. Departing from existing computationally expensive and slow threshold-based parametric control solutions, a novel computationally efficient and robust event-driven receding horizon control solution is proposed. Ongoing work is aimed to combine time-driven features of parametric control strategies with the proposed ED-RHC approach to construct a hybrid optimal control solution to the PMG problem.

REFERENCES

- [1] J. Trevathan and R. Johnstone, "Smart Environmental Monitoring and Assessment Technologies (SEMAT)—A New Paradigm for Low-Cost, Remote Aquatic Environmental Monitoring," *Sensors (Switzerland)*, vol. 18, no. 7, 2018.
- [2] K. Leahy, D. Zhou, C. I. Vasile, K. Oikonomopoulos, M. Schwager, and C. Belta, "Persistent Surveillance for Unmanned Aerial Vehicles Subject to Charging and Temporal Logic Constraints," *Autonomous Robots*, vol. 40, no. 8, pp. 1363–1378, 2016.
- [3] X. Meng, A. Houshmand, and C. G. Cassandras, "Multi-Agent Coverage Control with Energy Depletion and Repletion," in *Proc. of 58th IEEE Conf. on Decision and Control*, 2019, pp. 2101–2106.
- [4] S. L. Smith, M. Schwager, and D. Rus, "Persistent Monitoring of Changing Environments Using a Robot with Limited Range Sensing," in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2011, pp. 5448–5455.
- [5] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Optimal Threshold-Based Distributed Control Policies for Persistent Monitoring on Graphs," in *Proc. of American Control Conf.*, 2019, pp. 2030–2035.
- [6] S. Welikala and C. G. Cassandras, "Asymptotic Analysis for Greedy Initialization of Threshold-Based Distributed Optimization of Persistent Monitoring on Graphs," in *Proc. of 21st IFAC World Congress*, 2020.
- [7] 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 Trans. on Automatic Control*, vol. 63, no. 12, pp. 4204–4217, 2018.
- [8] X. Lin and C. G. Cassandras, "An Optimal Control Approach to The Multi-Agent Persistent Monitoring Problem in Two-Dimensional Spaces," *IEEE Trans. on Automatic Control*, vol. 60, no. 6, pp. 1659–1664, 2015.
- [9] Y. Khazaeni and C. G. Cassandras, "Event-Driven Trajectory Optimization for Data Harvesting in Multi-Agent Systems," *IEEE Trans. on Control of Network Systems*, vol. 5, no. 3, pp. 1335–1348, 2018.
- [10] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation Analysis and Optimization of Stochastic Hybrid Systems," *European Journal of Control*, vol. 16, no. 6, pp. 642–661, 2010.
- [11] W. Li and C. G. Cassandras, "A Cooperative Receding Horizon Controller for Multi-Vehicle Uncertain Environments," *IEEE Trans. on Automatic Control*, vol. 51, no. 2, pp. 242–257, 2006.
- [12] R. Chen and C. G. Cassandras, "Optimization of Ride Sharing Systems Using Event-driven Receding Horizon Control," in *Proc. of 2020 Intl. Workshop on Discrete Event Systems*, 2020.
- [13] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and Christos G. Cassandras, "Multi-Agent Infinite Horizon Persistent Monitoring of Targets with Uncertain States in Multi-Dimensional Environments," in *Proc. of 21st IFAC World Congress*, 2020.
- [14] J. Yu, M. Schwager, and D. Rus, "Correlated Orienteering Problem and its Application to Persistent Monitoring Tasks," *IEEE Trans. on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.
- [15] S. Welikala and C. G. Cassandras, "Event-Driven Receding Horizon Control For On-line Distributed Persistent Monitoring on Graphs," 2020. [Online]. Available: <http://arxiv.org/abs/2003.11713>