2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Bilevel Optimization Scheme for Persistent Monitoring

Jonas Hall[1], Logan E. Beaver[1], Christos G. Cassandras[1,2], Sean B. Andersson[1,3]

***Abstract*— In this paper we study an infinite-horizon persistent monitoring problem in a two-dimensional mission space containing a finite number of statically placed targets. At each target we assume a constant accumulation of uncertainty, which the agent is capable of reducing by taking local measurements with an onboard sensor. We derive a steady-state minimum time periodic trajectory over which each target uncertainty is driven to zero at least once. A hierarchical decomposition leads to purely local optimal control problems, coupled via boundary conditions. We optimize the local trajectory segments as well as the boundary conditions in an on-line bilevel optimization scheme.**

I. Introduction

Monitoring a dynamically changing environment in an efficient and cost-feasible manner has long attracted attention due to its broad applicability to areas such as ocean monitoring [1], forest monitoring [2], wildfire surveillance [3], data harvesting [4], or particle tracking [5]. A common approach is to place static sensors in order to maximize the monitored area or to maximize event detection probability, which in the literature is known as the coverage control problem [6]. However, employing a large number of static sensors can be expensive and inflexible. Hardware and software advances have enabled the replacement of static sensors with mobile autonomous agents equipped with sensors. The coverage control problem was thus extended to the Persistent Monitoring (PM) problem [7].

Over the last decade this problem has accumulated a rich set of formulations and variations. For some formulations the dynamic environment consists of a connected and typically compact subset of $\mathbb{R}^n$. In this setting the agents are often tasked to detect rogue elements appearing at unknown locations [8], or to minimize the cumulative average value of a dynamically changing scalar field [9]. Other formulations, as is the case in this paper, focus on a finite set of targets within the environment. Typical tasks then consist of detecting stochastic events at known locations [10], or minimizing the maximum revisit time along a periodic trajectory [11]. Usually the targets are spatially static, however, some formulations consider mobile targets as well [12], [13], [14].

A common subproblem of PM tasks consists of determining a periodic visiting sequence of all the targets, which in and of itself is NP-hard since it is more general than the Traveling Salesperson Problem (TSP), due to the dynamic nature of the problem. Even if a good visiting sequence is determined, computing optimal agent trajectories (with respect to a given metric such as minimum time or minimum energy) remains challenging. In order to monitor a given target we require the agent to be close to it. However, the more time the agent spends monitoring one target, the more cost is accumulated at all other targets. On the other hand, if the agent moves too quickly past a target, then the local cost, and thus also the global cost, is insufficiently reduced. A challenge in designing trajectories is to manage this trade-off. Due to the difficulties of solving PM problems, they are often decomposed and many contributions focus on specific subproblems. One such decomposition is the path-velocity decomposition [15]. However, this decomposition is always suboptimal unless the agent's local sensing capability is independent of the target-agent distance. Examples for velocity controllers along a given path can be found in [16], [17]. The vast majority of methods for trajectory optimization work off-line [9], [18], though the authors of [19] introduced an on-line trajectory optimization approach. Inherently different to the approach of decomposition is that of abstraction. Such methods formulate the mission space using a graph topology, where each target is described as a node and edges between two targets reflect the travel cost between those targets [20], [21], [11]. Such methods aim at solving the target visiting sequence.

In this paper we consider a PM formulation with a single agent and M targets, each of which is associated with an internal state that models uncertainty. The goal is to minimize the infinite-horizon average uncertainty. We introduce a method that optimizes the agent's trajectory on-line. Similar to [18], we decompose the problem into purely local Optimal Control Problems (OCP), the solutions of which provide decoupled trajectory segments. We then solve these OCPs using a direct multiple shooting approach [22]. While modern solvers are able to treat optimal control problems with hybrid dynamics directly [23], [24], we utilize this decomposition for the simple reason that the dimension of the local problems become independent of the number of targets. The contributions of this paper are

This work was supported in part by NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, CNS-2149511, by AFOSR under FA9550-19-1-0158, by ARPA-E under DE-AR0001282, and by the MathWorks.

[1]Division of Systems Engineering, Boston University, USA
[2]Department of Electrical and Computer Engineering
[3]Division of Mechanical Engineering, Boston University, USA
{hallj, lebeaver, ...}@bu.edu

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

(i) Thm. 2, which shows the existence of an exact relaxation for the local OCP (5) with hybrid dynamics, and
(ii) A bilevel optimization scheme that optimizes the agent's trajectory on-line.

The remainder of this paper is organized as follows. Sec. II introduces the considered PM problem. In Sec. III we introduce the decomposition into two layers: a sequence planner on the higher level; and a low-level layer generating the individual trajectory segments. In Sec. IV we analyze the low-level problems in detail. We then utilize a gradient descent method in Sec. V in order to optimize the boundary conditions that are imposed on the lower levels. Sec. VI discusses results in a comparison to a greedy solution.

II. Problem Formulation

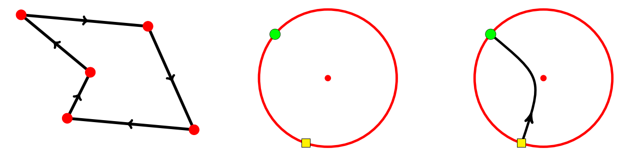
We are interested in a PM problem with a single agent and M targets indexed by $\mathcal{T} = \{1, 2, \dots, M\}$. We denote the agent's position by $s(t) \in \mathbb{R}^2$ which evolves according to the first-order system $\dot{s}(t) = u(t)$ with bounded control input $\|u(t)\| \leq 1$, where $\|\cdot\|$ denotes the Euclidean norm. The fixed positions of the targets are denoted by $x_1, x_2, \dots, x_M \in \mathbb{R}^2$. We assume that each target $i \in \mathcal{T}$ is associated with an internal state that models a measure of uncertainty $R_i \in \mathbb{R}_{\geq 0}$, which evolves according to the dynamics $\dot{R}_i = f_{R_i}$ given by

$$f_{R_i}(R_i, s) = \begin{cases} 0, & R_i = 0 \wedge A_i - B_i p_i(s) < 0, \\ A_i - B_i p_i(s), & \text{otherwise}, \end{cases}$$

where $B_i > A_i > 0$, and $p_i(s) = \max\left(0, 1 - \frac{\|s - x_i\|^2}{r_i^2}\right)$ is the monitoring model with sensing range $r_i > 0$. It is fairly straightforward to consider more complicated sensing functions so long as they remain monotonic in the agent-target distance.

We are interested in minimizing the average uncertainty over an infinite time horizon. While it is possible to characterize optimal solutions in one-dimensional settings [25], [26], [13], the problem becomes much harder in two-dimensional settings and such characterizations remain unknown. We thus rely on heuristics. Previous results [25], [26], [13] indicate that optimal trajectories typically drive the uncertainty of each target to zero before visiting another target. Additionally, we make the observation that in order to minimize a target's uncertainty accumulation during a trajectory segment at which it is not monitored is achieved by minimizing the duration of that time interval, since the uncertainty grows linearly in time with a rate of A_i. Motivated by these behaviors, we formulate the following persistent monitoring problem.

Problem *Find a steady-state minimum time periodic trajectory over which each target uncertainty is drained, i.e., driven down to zero, during each visit.*



(a) (b) (c)

Fig. 1: Illustrating the three optimization problems of the decomposition: (a) sequencing; (b) entrance and departure point optimization; and (c) local trajectory optimization.

Assumptions *Throughout this paper we assume that: 1) the initial uncertainties $R_i(0)$ are known for all targets $i \in \mathcal{T}$; 2) the sensing areas around the targets do not intersect; and 3) there exists a steady-state solution.*

These assumptions are typical in the given PM setting [16], [26], [13]. Note that assumption 2) is fundamental for the decomposition, whereas assumption 3) is fundamental for convergence. We remark that the existence of steady-state trajectories strongly depends on the topology of the mission space as well as the parameters A, B, and r. While the existence of steady-state solutions can be proven when the uncertainty model is replaced by a Kalman filter model [27], [18], this remains an open problem in the given setting.

III. Hierarchical Decomposition

With the problem set up, it is a natural task to identify characterizing properties of an optimal periodic trajectory. It is immediately evident that the agent is required to visit each of the targets in order to drive their uncertainties down to zero. This understanding directly induces a two-level hierarchy: a higher level with the objective of finding a target visiting sequence; and a lower level of steering the agent so as to 1) satisfy the target visiting sequence and 2) *drain* each of the target uncertainties, i.e., drive them down to zero. We will formulate the low-level problems as OCPs and we refer to those problems as the *local OCPs*, since solving them only requires knowledge of local information of the visited target. In order to connect the two levels, we introduce a coordinator[1], which takes a visiting sequence and then coordinates the local trajectories by providing the boundary conditions of the local OCPs. Additionally, it is the coordinator's task to optimize those boundary conditions on-line.

The optimization problems within the individual levels are depicted in Fig. 1: 1a illustrates the problem of finding a target visiting sequence; 1b illustrates the problem of optimizing the entrance (yellow square) and departure (green circle) points when visiting a target;

[1]The coordinator is not to be confused with coordinators in multi-agent systems, which coordinate information between agents. Here it coordinates information between trajectory segments.

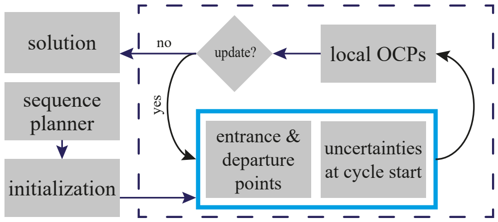


Fig. 2: Illustrating the workflow of the coordinator, which is part of the agent's control system.

and 1c depicts a local OCP, which determines the trajectory that drives the target uncertainty to zero while respecting the boundary conditions from 1b. Note that the red circle in 1b and 1c depicts the sensing radius around the considered target.

In this paper we focus on the optimization of the entrance and departure points together with the low-level trajectories. However, we remark that visiting sequences can be computed using a graph-based abstraction, e.g., via TSP or other methods specifically designed for PM problems [21], [11].

Coordinating the local trajectories. From here on we assume that a periodic target visiting sequence $\{i_1, i_2, \dots i_K, i_1\}$ is provided. The coordinator is tasked to realize the visiting sequence and coordinate the requirement of driving the target uncertainties down to zero. To do this, we note that during the kth target visit, the agent begins sensing the target i_k at a specific point in space, which we denote by s_k^{φ} and refer to it as the entrance point (yellow square in 1b). Similarly, there exists a departure point s_k^{ψ} (green circle in 1b), i.e., a point at which the agent last sensed the target. The coordinator passes the entrance and departure points down to the local OCP solver generating the local trajectory segments as discussed at the end of this section. In return, the local OCPs provide dual variables specifying the cost associated with the entrance and departure constraints. The coordinator then utilizes these dual variables to optimize the entrance and departure points with the goal of minimizing the total cycle time (see Sec. V).

Fig. 2 depicts the proposed workflow of the coordinator (dashed box). It receives an initial guess of the entrance and departure points generated from the visiting sequence (the generation is discussed in Sec. V). The coordinator then calls the local OCP solver in an event-based fashion, i.e., whenever the agent starts or stops sensing a target. On completion of a cycle, the coordinator updates the entrance/departure points as well as the target uncertainties at cycle start, or terminates the algorithm if the cost gradients with respect to the entrance and departure points are sufficiently small and steady-state is reached.

Formulating the local OCPs. There are two types of local OCPs to be solved: that of driving a target's uncertainty to zero (*draining problem*) and that of moving from the departure point of one target s_k^{ψ} to the entrance point of the next target s_{k+1}^{φ} (*switching problem*).

Given an unconstrained environment and first-order dynamics, the switching problem becomes trivial as it is given by a maximal and constant control input that moves the agent from one point to another along a straight line. However, the current formulation is capable of adapting to other scenarios. For example, if obstacles are present in the environment then the problem becomes more complicated but can often still be solved using optimal control techniques [28]. We solely require the local switching problem to provide cost sensitivities with respect to the constraints that fix the boundary conditions s_k^{ψ} and s_{k+1}^{φ}.

Let us now focus on the draining problem, which consists of finding a time optimal trajectory that drives the target uncertainty down to zero while satisfying the constraints imposed by the coordinator. Specifically, this is given by the OCP

$$\min_{u(\cdot), T_k} \quad \int_0^{T_k} \mathrm{d}t \tag{1a}$$

$$\text{s.t.} \quad \dot{s}(t) = u(t), \tag{1b}$$

$$\dot{R}_{i_k}(t) = f_{R_{i_k}}(R_{i_k}(t), s(t)), \tag{1c}$$

$$\|u(t)\|^2 \leq 1, \tag{1d}$$

$$\min_{\tau \in [0, T_k]} R_{i_k}(\tau) = 0, \tag{1e}$$

$$s(0) = s_k^{\varphi}, \tag{1f}$$

$$s(T_k) = s_k^{\psi}, \tag{1g}$$

$$R_{i_k}(0) = \check{R}_k, \tag{1h}$$

where s_k^{φ} and s_k^{ψ} denote the respective entrance and departure points, and $\check{R}_k$ denotes the uncertainty at arrival time, all of which are passed down from the coordinator. At first glance, this problem seems challenging to solve due to the non-smoothness of $f_{R_{i_k}}$ and the unconventional constraint (1e), which enforces the uncertainty to be drained. In the next section we discuss how the problem can be solved efficiently.

IV. Solving the Draining OCP

In this section, we devote our attention to the local draining OCP (1), the main result being Thm. 2, which shows the existence of an exact relaxation. Let us denote by δ_{i_k} the radius of the zero-level set of the function $f_{R_{i_k}}$ as depicted in Fig. 3. We begin with a theorem that characterizes the final segment of the trajectory. Proofs to all statements are provided in the preprint [29].

Theorem 1 *For any optimal trajectory of (1) there exists a unique time t_k^0 such that $\|s^*(t_k^0) - x_{i_k}\| = \delta_{i_k}$ and $R_{i_k}^*(t_k^0) = 0$. Further, it holds*

$$u^*(t) = \frac{s_k^{\psi} - s^*(t_k^0)}{\|s_k^{\psi} - s^*(t_k^0)\|} \tag{2}$$

for every $t \in [t_k^0, T_k^*]$.

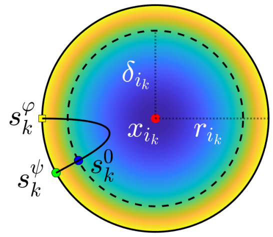


Fig. 3: Illustrating a solution of the draining OCP (1) of target x_{i_k} with entrance point s_k^φ, outer departure point s_k^ψ and inner departure point s_k^0 (cf. (5)). The function $f_{R_{i_k}}$ takes positive values when the agent-target distance exceeds δ_{i_k}, whereas it is negative in the interior of the inner circle (dashed).

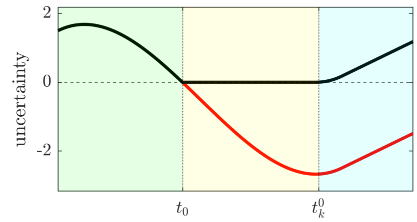


Fig. 4: Projecting the relaxation (red) to zero during the yellow interval, and shifting it during the blue interval by the violation at t_k^0 recovers the true uncertainty trajectory (black).

A special case of the optimal control occurs whenever the initial uncertainty $\check{R}_{i_k}$ is large enough such that the greedy control policy

$$\tilde{u}(t) = \begin{cases} \frac{x_{i_k} - s_k^\varphi}{r_{i_k}}, & \text{if } 0 \le t < r_{i_k}, \\ 0, & \text{if } r_{i_k} \le t < T_k^* - r_{i_k}, \\ \frac{s_k^\psi - x_{i_k}}{r_{i_k}}, & \text{if } T_k^* - r_{i_k} \le t \le T_k^*, \end{cases} \quad (3)$$

becomes optimal, as captured in the next statement.

Proposition 1 *If*

$$\check{R}_{i_k} \ge -\left(A_{i_k} - \tfrac{2}{3} B_{i_k}\right) r_{i_k} - \delta_{i_k}(A_{i_k} - B_{i_k}) - \frac{\delta_{i_k}^3}{3 r_{i_k}^3} B_{i_k} \quad (4)$$

then the greedy control policy (3) is an optimal control of (1).

Finally, we provide a statement that reformulates (1) into the smooth OCP

$$\min_{u(\cdot),\, t_k^0,\, s_k^0} \quad \int_0^{t_k^0} \mathrm{d}t + \|s_k^\psi - s_k^0\| \quad (5a)$$
$$\text{s.t.} \quad \dot{s}(t) = u(t), \quad (5b)$$
$$\dot{R}_{i_k}(t) = A_{i_k} - B_{i_k} p_{i_k}(s(t)), \quad (5c)$$
$$\|u(t)\|^2 \le 1, \quad (5d)$$
$$R_{i_k}(t_k^0) \le 0, \quad (5e)$$
$$s(0) = s_k^\varphi, \quad (5f)$$
$$s(t_k^0) = s_k^0, \quad (5g)$$
$$\|s_k^0 - x_{i_k}\| = \delta_{i_k}, \quad (5h)$$
$$R_{i_k}(0) = \check{R}_k. \quad (5i)$$

Theorem 2 *The relaxation (5) is exact, i.e., any optimal trajectory s^* of OCP (5) is also optimal for (1). Further, the respective uncertainty trajectory can be recovered from the relaxed counterpart.*

V. Optimizing the Local Boundary Conditions

The goal of this section is to minimize the cycle time T given a sequence of target visits $i_1, i_2, \ldots, i_K \in \mathcal{T}$, where K denotes the length of the visiting sequence. By decomposing the cycle into its local segments, we can express the total cycle time in terms of the entrance and departure points, i.e., as

$$T = \sum_{k=1}^{K} T_k^*(s_k^\varphi, s_k^0) + \Delta_k^*(s_k^0, s_{k+1}^\varphi),$$

where T_k^* denotes the time of the kth draining trajectory, and $\Delta_k^* = \|s_k^0 - s_{k+1}^\varphi\|$ denotes the kth switching time, i.e., the time taken from the (inner) departure point s_k^0 to the next entrance point s_{k+1}^φ. Now note that the entrance and departure points can be expressed as

$$s_k^\varphi = x_{i_k} + r_{i_k} \begin{pmatrix} \cos(\varphi_k) \\ \sin(\varphi_k) \end{pmatrix} \quad \text{and} \quad s_k^0 = x_{i_k} + \delta_{i_k} \begin{pmatrix} \cos(\psi_k) \\ \sin(\psi_k) \end{pmatrix},$$

respectively, where φ_k and ψ_k correspond to the polar coordinate angles of the entrance and departure points in the coordinate system fixed upon the target i_k. Motivated by Thm. 2, we choose to optimize the inner departure point instead of the outer departure point. Then the only degrees of freedom are the angles φ_k and ψ_k. From here on we denote by $\varphi, \psi \in \mathbb{R}^K$ the vectors that contain the respective entrance and departure angles. We may then express the total cycle time $T(\varphi, \psi)$ as a function of these parameters. We are left to solve the unconstrained bilevel minimization problem

$$\min_{\varphi, \psi} \quad \sum_{k=1}^{K} T_k^*(\varphi_k, \psi_k) + \Delta_k^*(\psi_k, \varphi_{k+1}). \quad (6)$$

Solving this problem can be done on-line in the following manner. We first initialize the entrance and departure points using

$$\psi_k = \operatorname{atan2}(\vartheta_y^k, \vartheta_x^k), \;\; \varphi_{k+1} = \operatorname{atan2}(-\vartheta_y^k, -\vartheta_x^k), \quad (7)$$

where $\vartheta^k = x_{k+1} - x_k$. This natural initialization places s_k^0 and s_{k+1}^φ on the intersection of the straight line from target i_k to i_{k+1} with the respective sensing range circles. We place the agent at the exit point of target i_K and apply the constant control $(s_1^\varphi - s_K^0)/\Delta_K$. When the agent arrives at the first entrance point, we solve a discretized version of the smooth draining OCP (5) for the first target (the discretization is discussed in Sec. VI). Note that this provides an open loop control law during the draining period. Alternatively we could apply a closed loop controller to track the computed trajectory, or close the loop using Model Predictive Control (MPC) [30]. When reaching the departure point, we repeat the process for the next target.

On completion of a cycle we compute the gradients

$$\frac{\partial T}{\partial \varphi_k} = \frac{\partial T_k^*}{\partial \varphi_k} + \frac{\partial \Delta_{k-1}^*}{\partial \varphi_k}, \quad \frac{\partial T}{\partial \psi_k} = \frac{\partial T_k^*}{\partial \psi_k} + \frac{\partial \Delta_k^*}{\partial \psi_k}, \tag{8}$$

of the cycle time $T(\varphi, \psi)$. Note that gradients of Δ^* can be computed analytically, while evaluation of the gradients of T_k^* can be done by using the dual variables — or Lagrange multipliers [31] — of the respective constraints. Let us denote by $\lambda_{\varphi_k}, \lambda_{\psi_k} \in \mathbb{R}^2$ the dual variables of the entrance (5f) and departure constraints constraint (5g) of the kth local OCP, respectively. Applying the chain rule then provides

$$\begin{aligned} \frac{\partial T_k^*}{\partial \varphi_k} &= \lambda_{\varphi_k}^\top r_{i_k} \begin{pmatrix} -\sin(\varphi_k) \\ \cos(\varphi_k) \end{pmatrix}, \\ \frac{\partial T_k^*}{\partial \psi_k} &= \lambda_{\psi_k}^\top \delta_{i_k} \begin{pmatrix} -\sin(\psi_k) \\ \cos(\psi_k) \end{pmatrix}. \end{aligned}$$

We then update the parameters using a simple gradient descent law

$$\varphi_k \leftarrow \varphi_k - \alpha \frac{\partial T}{\partial \varphi_k}, \quad \psi_k \leftarrow \psi_k - \alpha \frac{\partial T}{\partial \psi_k}, \tag{9}$$

where α is chosen using a diminishing step-size rule.

VI. Numerical Results

Motivated by the fact that greedy policies are able to produce optimal solutions under certain scenarios (see Prop. 1), we compare the proposed method to the greedy control policy: move towards the target (and potentially dwell there) until uncertainty hits 0, then move to the next target. We consider homogeneous targets with $A_i = 1$, $B_i = 20$, $r_i = 3$, and $R_i(0) = 0$.

We discretize the local OCPs via direct multiple shooting [22] using explicit Euler integration over 20 nodes. We model this in Matlab via CasADi [32] and then solve the resulting nonlinear programs using IPOPT [33]. The underlying hardware consists of an Intel i5 processor running at 1.60GHz with 16GB of RAM.

Fig. 5 shows that even though the trajectories appear to be similar, our approach shows a 20% reduction in total travel time over the greedy policy, as can be seen on the right plot depicting the respectively obtained steady-state cycles with periods of 41.8 and 52.3 time units. Furthermore, the proposed method leads to a smooth control profile, which is favorable for tracking feasibility. We conjecture that this smoothness is a fundamental behavior for solutions of (6), with exceptions being settings where the optimal entrance and departure points coincide, or settings where the the initial uncertainty is large enough to satisfy (4). The following intuition justifies this conjecture: if the entrance (or departure) transition is non-smooth, then this indicates that the angle between the entrance and departure points is too large. Reducing the angle between the entrance and departure point always reduces the local draining time T_k^*. The only thing that prevents the entrance and departure points from converging to each other is the trade-off introduced by the potentially increased switching times Δ_{k-1} and Δ_k. The equilibrium of this trade-off, namely the solution of (6), may induce a natural property of smooth transitions.

We now shift our attention to the computational efficiency of the proposed method. The right plot in Fig. 5 shows the CPU timings for this experiment, where we recall that the draining OCP refers to (5) and the switching OCP refers to the (trivial) problem of switching from target's departure point to the next target's entrance point. The relaxation of the hybrid dynamics as well as the reduction of the state space dimensionality lead to trajectory segments computed in fractions of a second, suggesting real-time feasibility for systems with update rates of 50-80 Hz. Fig. 6 depicts the convergence and demonstrates robustness towards the initial conditions.

VII. Conclusion and Future Work

In this paper we considered a two-dimensional infinite-horizon PM problem, in which we are interested in finding a minimum time draining cycle. By decomposing the problem into local OCPs on the lowest level, and coordinating their trajectories via higher level parameters, we were able to prove the existence of an exact relaxation for the underlying hybrid dynamics. These two layers are coupled within a bilevel optimization scheme, with which the agent's trajectory is optimized on-line.

In future work we aim at analyzing the conjecture made in Sec. VI, as well as extending to scenarios to three dimensions, or to multi-agent settings. Furthermore, a particularly interesting extension could arise when the draining condition is relaxed. This could be done by introducing new parameters that specify the right hand side of (1e), which could then be optimized in a similar way as was done here with the entrance and departure points. Apart from extending the introduced approach, we also desire comparing the proposed approach to existing methods, e.g., by utilizing the general optimal control problem solver [23], or to the method suggested in [18].

References

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

[2] S. Naderi, K. Bundy, T. Whitney, A. Abedi, A. Weiskittel, and A. Contosta, "Sharing wireless spectrum in the forest ecosystems using artificial intelligence and machine learning," *International Journal of Wireless Information Networks*, vol. 29, no. 3, pp. 257–268, 2022.

[3] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[4] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi, "Efficient data harvesting in mobile sensor platforms," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pp. 5–pp, IEEE, 2006.

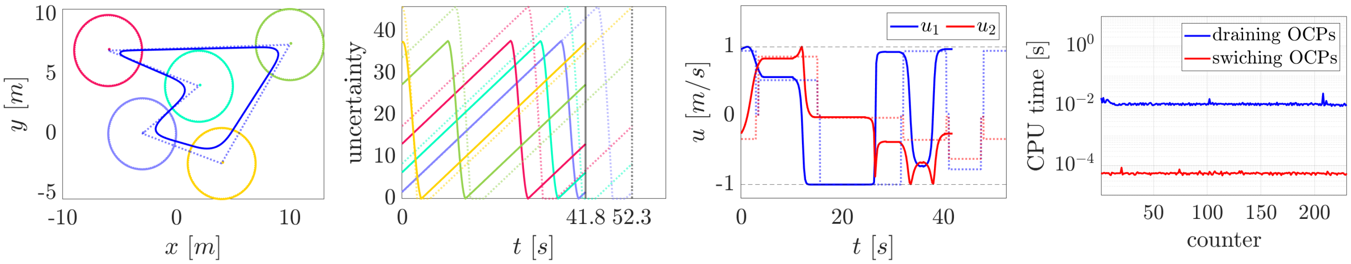


Fig. 5: Comparing the greedy solution (dotted) to the solution obtained via the proposed method (solid).

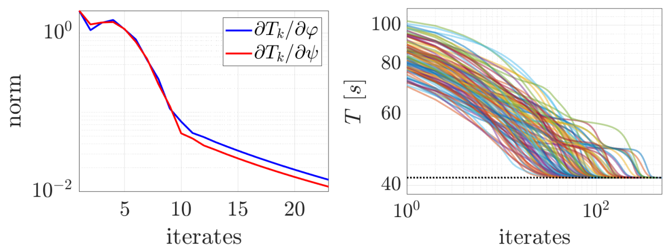


Fig. 6: Depicting the convergence of the gradient norm (left) and the convergence of 100 randomly perturbed initializations to the same optimizers (right).

[5] S. C. Pinto, N. A. Vickers, F. Sharifi, and S. B. Andersson, "Tracking multiple diffusing particles using information optimal control," in *2021 American Control Conference (ACC)*, pp. 4033–4038, IEEE, 2021.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[7] C. G. Cassandras, X. C. Ding, and X. Lin, "An optimal control approach for the persistent monitoring problem," in *2011 50th IEEE conference on decision and control and european control conference*, pp. 2907–2912, IEEE, 2011.

[8] M. Boldrer, L. Lyons, L. Palopoli, D. Fontanelli, and L. Ferranti, "Time-inverted kuramoto model meets lissajous curves: Multi-robot persistent monitoring and target detection," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 240–247, 2022.

[9] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1659–1664, 2014.

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

[11] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Optimal UAV route planning for persistent monitoring missions," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 550–566, 2020.

[12] R. P. Anderson and D. Milutinović, "A stochastic approach to dubins vehicle tracking problems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2801–2806, 2014.

[13] J. Hall, S. B. Andersson, and C. G. Cassandras, "Optimal persistent monitoring of mobile targets in one dimension," *arXiv preprint arXiv:2210.01294*, 2022.

[14] Y. Wang, Y. Wang, Y. Cao, and G. Sartoretti, "Spatio-temporal attention network for persistent monitoring of multiple mobile targets," *arXiv preprint arXiv:2303.06350*, 2023.

[15] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The international journal of robotics research*, vol. 5, no. 3, pp. 72–89, 1986.

[16] S. L. Smith, M. Schwager, and D. Rus, "Persistent monitoring of changing environments using a robot with limited range sensing," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5448–5455, IEEE, 2011.

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

[18] M. Ostertag, N. Atanasov, and T. Rosing, "Trajectory planning and optimization for minimizing uncertainty in persistent monitoring applications," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 1, pp. 1–19, 2022.

[19] G. Notomista, C. Pacchierotti, and P. R. Giordano, "Online robot trajectory optimization for persistent environmental monitoring," *IEEE Control Systems Letters*, vol. 6, pp. 1472–1477, 2021.

[20] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 138–154, 2014.

[21] S. Welikala and C. G. Cassandras, "Event-driven receding horizon control for distributed persistent monitoring in network systems," *Automatica*, vol. 127, p. 109519, 2021.

[22] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[23] A. Nurkanović and M. Diehl, "NOSNOC: A software package for numerical optimal control of nonsmooth systems," *IEEE Control Systems Letters*, 2022.

[24] J. Hall, A. Nurkanović, F. Messerer, and M. Diehl, "A sequential convex programming approach to solving quadratic programs and optimal control problems with linear complementarity constraints," *IEEE Control Systems Letters*, vol. 6, pp. 536–541, 2021.

[25] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2012.

[26] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal event-driven multiagent persistent monitoring of a finite set of data sources," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4204–4217, 2018.

[27] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Multi-agent persistent monitoring of targets with uncertain states," *IEEE Transactions on Automatic Control*, 2022.

[28] L. E. Beaver, R. Tron, and C. G. Cassandras, "A graph-based approach to generate energy-optimal robot trajectories in polygonal environments," *IFAC-PapersOnLine (to appear)*, 2023. 22nd IFAC World Congress (to appear).

[29] J. Hall, L. E. Beaver, C. G. Cassandras, and S. B. Andersson, "A bilevel optimization scheme for persistent monitoring," *arXiv preprint arXiv:2304.03667*, 2023.

[30] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE control systems magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[31] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[32] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[33] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.