A Semidefinite Programming Approach to Discrete-time Infinite Horizon Persistent Monitoring

Samuel C. Pinto[1], Sean B. Andersson[1,2], Julien M. Hendrickx[3], and Christos G. Cassandras[2,4]

[1]Dept. of Mechanical Engineering, [2]Division of Systems Engineering,
[4]Dept. of Electrical and Computer Engineering
Boston University, Boston, MA 02215, USA
[3]ICTEAM Institute, UCLouvain, Louvain-la-Neuve 1348, Belgium
{samcerq,sanderss,cgc}@bu.edu, julien.hendrickx@uclouvain.be

Abstract— We investigate the problem of persistent monitoring, where a mobile agent has to survey multiple targets in an environment in order to estimate their internal states. These internal states evolve with linear stochastic dynamics and the agent can observe them with a linear observation model. However, the signal to noise ratio is a monotonically decreasing function of the distance between the agent and the target. The goal is to minimize the uncertainty in the state estimates over the infinite horizon. We show that, for a periodic trajectory with fixed cycle length, the problem can be formulated as a set of semidefinite programs. We design a scheme that leverages the spatial configuration of the targets to guide the search over this set of optimization problems in order to provide efficient trajectories. Results are compared to a state of the art approach and we obtain improvements of up to 91% in terms of cost in a simple scenario, with much lower computational time.

I. INTRODUCTION

We study the problem of persistent monitoring of a finite set of targets by a mobile agent, where each target has an internal state that evolves over time with some degree of uncertainty in its dynamics. The agent can observe these internal states when it is close to a given target, however, in order to monitor all the targets, it needs to move around the environment and visit them infinitely often. This paradigm finds applications in many domains, such as ocean temperature monitoring [1] and pipeline inspection [2]. This problem is closely related to the Traveling Salesman Problem (TSP), where, given a set of targets, the goal is to find a cycle in which the agents efficiently visit all the targets in order to minimize the traveled distance or total travel time. The major difference between the TSP and the problem we are dealing with in this paper is that the optimization goal we consider is to minimize the uncertainty rather than distance. We emphasize that due to the dynamic nature of uncertainty, we cannot model the agent as being embedded in a graph with fixed cost on the edges, as assumed in TSP.

In the realm of persistent monitoring, significant previous work has been done. In [1], a variant of the Rapidly

Exploring Random Trees (RRT) algorithm denoted Rapid Random Cycles (RRC) was designed for cyclic discrete time persistent monitoring while [3] proposed an optimal control approach for the continuous time version of the persistent monitoring problem that relied on a solution of the two-point boundary value problem resulting from a Hamiltonian analysis. However, the solution of the two-point boundary value problem is computationally expensive.

We have previously studied this problem using the assumption that the internal state of the target evolves according to linear, stochastic dynamics with linear observations, but corrupted with noise [4]. However, these works approached the problem from a continuous-time perspective. The trajectory of the agents was parameterized using a finite number of parameters, and then gradient descent was used to optimize over these parameters, leading to scalable solutions. The trajectories often converged to local optima and provided no insight on how far from global optimality the solutions were. To the best of the authors' knowledge, the only work that includes a notion of global (asymptotic probabilistic) optimality in persistent monitoring of uncertain states is [1] and traditional tools with global guarantees, such as dynamic programming, are computationally prohibitive even for a small number of targets. In this work, we intend to have more than a local notion of optimality while retaining the ability to generate efficient trajectories for problems with a small to moderate number of targets within a reasonable computational time.

In this paper, however, we limit ourselves to a single agent and a discrete time model and formulate the problem in a way that both the computation of the steady state uncertainty and the local optimization of the trajectory can be framed as a single optimization problem, a semidefinite program (SDP). The formulation and solution of this joint optimization problem is the main contribution of this paper. This contrasts with our previous approach [4] where, in a gradient descent scheme, the steady state uncertainty had to be computed through a computationally costly algorithm. In the present work we benefit from efficient and reliable SDP solvers and are able to quickly solve a local version of the persistent monitoring problem. Moreover, we are not limited to local optimality, as we have also embedded this

local SDP-based optimizer into a higher level algorithm that searches globally for different periodic trajectories. This higher level scheme leverages the spatial distribution of the targets and then feeds the lower level optimization with configurations that will lead to feasible schedules. Due to the infinite number of candidate trajectories, we still are not able to guarantee global optimality. However, simulation results show that the approach proposed in this paper is able to efficiently handle problems with a small number of targets, providing trajectories with good performance even in the initial iterations of the higher level algorithm. Moreover, trajectories generated with the approach here proposed give a significant reduction (91%) in terms of estimation error when compared to RRC [1], while also showing significant computational time reduction.

II. Problem Formulation

Consider an environment with N target locations that should be monitored. Each target i has an internal state $\phi_i \in \mathbb{R}^{L_i}$ that continuously evolves over time. An agent can move and sense these targets. We assume that the dynamics of the targets and observations of its internal state by the agent are given according to the following discrete-time model:

$$\phi_i(k+1) = A_i\phi_i(k) + w_i(k), \tag{1a}$$

$$z_i(k) = H_i(k)\phi_i(k) + v_i(k), \tag{1b}$$

where $w_i(k)$ and $v_i(k)$ are zero mean, mutually independent white Gaussian processes with constant covariance matrices Q_i and R_i, respectively.

Assume that at a given time step k, the agent has a position $s(k)$ and that the agent can deterministically control its position according to the following model

$$s(k+1) = s(k) + u_k, \ \|u_k\| \le u_{\max}. \tag{2}$$

We note that the choice of these simple dynamics is made for ease of presentation. Extending to more complicated agent dynamics is straightforward. Motivated by the fact that real sensors (such as sonar, cameras and lidars) normally have a finite range and that their observation quality usually decays as the agent moves farther from the sensing target, we assume the following model for $H_i(k)$:

$$H_i(k) = \sqrt{\gamma_i(k)}H_{i,\max},$$
$$\gamma_i(k) = \begin{cases} \left(1 - \frac{(s(k)-x_i)^2}{r_i^2}\right), & \|s(k)-x_i\| \le r_i, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where x_i is the position of target i and $H_{i,\max}$ is a matrix.

This particular structure of $H_i(k)$ captures the fact that the power of the signal decays as the agent moves farther from the target, while the noise power stays constant. If the distance between agent and target is larger than r_i, the intensity of the signal is zero, which is equivalent to not sensing at all. The particular quadratic decay was chosen due to the fact that it can be easily incorporated into an SDP. We plan to study extensions to other shapes in future work.

When estimating the states, the goal is to find an unbiased estimator $\hat{\phi}_i(k)$ that minimizes the mean squared estimation error over an infinite time horizon. Letting $\Sigma_i(k)$ denote the covariance matrix of the estimator $\hat{\phi}_i(k)$ and noting that $E[(\hat{\phi}_i - \phi_i)^T(\hat{\phi}_i - \phi_i)] = \text{tr}(\Sigma_i)$, we define the cost C of a particular infinite horizon trajectory of the agent as:

$$C = \lim_{M \to \infty} \frac{1}{M} \sum_{j=1}^{M} \sum_{i=1}^{N} \text{tr}(\Sigma_i(j)) \tag{4}$$

when the above limit exists. When it does not exist, the cost is defined as infinity.

Since both the dynamics and the observations are linear and the process and observation noises are Gaussian and uncorrelated, with the additional assumption that the distribution of $\phi_i(0)$ is Gaussian and uncorrelated with $w_i(k)$ and $v_i(k)$, we know that the optimal estimator is a Kalman Filter [5]. Therefore, in order to efficiently approach the persistent monitoring problem, we only need to design a trajectory that will influence the sensing matrices $H_i(k)$ in order to reduce the overall uncertainty. Note that $H_i(k)$ itself is directly related to the covariance matrix through the Kalman filter propagation equations.

In the framework of persistent monitoring where targets have to be visited infinitely often, a very natural assumption is that targets will follow a periodic schedule, as already exploited in some of our previous work on the continuous time version of the problem [4], [6]. Periodic schedules can approximate arbitrarily well the cost of an optimal schedule [7] and they naturally provide an upper bound to the inter-visit time of targets. One particularly interesting property of periodic schedules is that, under some very natural assumptions, the covariance matrix of each of the targets converges to a steady state periodic solution. Therefore, we only need to design one period of the agent trajectory in order optimize the performance of the system over long horizons.

We make the following assumption that ensures that the entire internal state can indeed be estimated from the observations:

Assumption 1: The pair $(A_i, H_{i,\max})$ is observable, for every $i \in \{1, ..., M\}$.

Under this assumption, it can be shown that, if every target is visited at least once in a period, the covariance matrix will converge to a unique globally attractive steady state solution for any initial conditions $\Sigma_i(0)$, even if the internal state dynamics, captured by the matrix A_i, are unstable (the proof is a discrete time version of Prop. 2 in [6], and is omitted here due to space limitations). Therefore, assuming a cyclic trajectory of period τ where all the targets are visited,

$$C = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{N} \text{tr}(\Sigma_i(m)) = \frac{1}{\tau} \sum_{k=1}^{\tau} \sum_{i=1}^{N} \text{tr}(\bar{\Sigma}_i(k)) \tag{5}$$

where $\bar{\Sigma}_i(k)$ is the steady state covariance matrix and the indexes i and k refer respectively to the index of the target and index of the time step within the cycle.

III. COMPUTATION OF THE INFINITE HORIZON COST

In order to jointly optimize the trajectory and compute the infinite horizon cost using SDPs, we write the Kalman filter in a different format, known as the information filter. We first briefly recall the well known relationship between the Kalman filter and the information filter. Then we develop a scheme to compute the infinite horizon cost of a periodic trajectory using an SDP whose optimal solution satisfies the information filter equations.

A. Information Filter

Recall that the Kalman filter equations can be written in two steps (prediction and update) [5]. The covariance update in the prediction is given by

$$\Sigma_i(k|k-1) = A_i\Sigma_i(k-1|k-1)A_i^T + Q_i \qquad (6)$$

and in the update step by

$$\Sigma_i(k|k) = \Sigma_i(k|k-1) - \Sigma_i(k|k-1)H_i^T(k) \\ \times (H_i(k)\Sigma_i(k|k-1)H_i^T(k) + R_i)^{-1}H_i(k)\Sigma_i(k|k-1) \qquad (7)$$

where $\Sigma_i(k|k) = \Sigma_i(k)$ and $\Sigma_i(k|k-1)$ are the covariances at time k using information up to time k and $k-1$ respectively. For details, see, e.g., [5], [8]. Before moving to the information filter, we state the following assumption:

Assumption 2: Q_i, R_i and the initial covariance matrix $\Sigma_i(0)$ are positive definite for every $i \in \{1, ..., N\}$.

Under this assumption, we know that the covariance matrices $\Sigma_i(k|k-1)$ and $\Sigma_i(k|k)$ are positive definite. Let us define $P_i(k|k-1) = \Sigma_i^{-1}(k|k-1)$ and $P_i(k|k) = \Sigma_i^{-1}(k|k)$. Using the matrix inversion lemma in the prediction step [5] yields

$$P_i(k|k-1) = Q_i^{-1} \\ - Q_i^{-1}A_i(Q_i^{-1} + A_i^T P_i(k-1|k-1)A_i)^{-1}A_i^T Q_i^{-1}. \qquad (8)$$

Analogously, using the matrix inversion lemma on the update step leads to

$$P_i(k|k) = P_i(k|k-1) + H_i^T(k)R_iH_i(k). \qquad (9)$$

Merging both steps, we get the following recursion:

$$P_i(k|k) = Q_i^{-1} + H_i^T(k)R_iH_i(k) \\ - Q_i^{-1}A_i(Q_i^{-1} + A_i^T P_i(k-1|k-1)A_i)^{-1}A_i^T Q_i^{-1}, \qquad (10)$$

which is the well known information filter recursion [8] that we will use from this point forward in the paper. The information filter is optimal, since it consists simply of a rearrangement of the Kalman filter equations, where instead of propagating directly the covariance, the inverse of the covariance is propagated.

B. Cyclic Schedules and the algebraic Ricatti equation

Equation (5) shows that the average steady state uncertainty over a single cycle is equal (in the limit) to the mean squared estimation error over a very long time horizon. Therefore, computing these steady state covariance matrices is an essential part of solving the persistent monitoring problem. In this subsection, we discuss one method for computing $\bar{\Sigma}_i(k)$. We pick this specific method due to the fact that it can be easily integrated in the SDP framework that we will explore in the next subsection. We take an approach similar to [9], where the steady state covariance is computed using a single augmented algebraic Ricatti equation (ARE). We point out that we cannot directly use the results in [9] because they use the Kalman filter in its standard form and not the information version. Therefore, in order to use an ARE to compute the steady state behavior of the system, we define the following augmented inverse covariance

$$\tilde{P}_{i,k} = \begin{bmatrix} \bar{P}_i(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{P}_i(k+\tau-1) \end{bmatrix}, \qquad (11)$$

with $\bar{P}_i(k) = \bar{\Sigma}_i^{-1}(k)$, and augmented parameters $\tilde{\Lambda}_i = \text{diag}(A_i, ..., A_i)$, $\tilde{\Psi}_i = \text{diag}(Q_i, ..., Q_i)$, $\tilde{H}_{i,k} = \text{diag}(H_i(k), ..., H_i(k+\tau-1))$, $\tilde{R}_i = \text{diag}(R_i, ..., R_i)$. The recursion in (10) can be rewritten as:

$$\tilde{P}_{i,k} = \tilde{\Psi}_i^{-1} + \tilde{H}_{i,k}^T\tilde{R}_i\tilde{H}_{i,k} \\ - \tilde{\Psi}_i^{-1}\tilde{\Lambda}_i(\tilde{\Psi}_i^{-1} + \tilde{\Lambda}_i^T\tilde{P}_{i,k-1}\tilde{\Lambda}_i)^{-1}\tilde{\Lambda}_i^T\tilde{\Psi}_i^{-1}. \qquad (12)$$

For ensuring periodicity, we require that $P_i(k+\tau) = P_i(k)$, therefore,

$$\tilde{P}_{i,k+1} = J\tilde{P}_{i,k}J^T, \qquad (13)$$

$$J = \begin{bmatrix} 0_{L_i \times (N-1)L_i} & I_{L_i \times L_i} \\ I_{(N-1)L_i \times (N-1)L_i} & 0_{(N-1)L_i \times L_i} \end{bmatrix}. \qquad (14)$$

Defining $\tilde{Q}_i^{-1} = (J^T\tilde{\Psi}_iJ)^{-1}$ and $\tilde{A}_i = J^{-T}\tilde{\Lambda}_i$ and substituting into (12) we get the following algebraic Riccati equation for computing $\tilde{P}_{i,k}$ for each of the targets i:

$$\tilde{Q}_i^{-1} - \tilde{P}_{i,k} + \tilde{H}_{i,k}^T\tilde{R}_i\tilde{H}_{i,k} \\ - \tilde{Q}_i^{-1}\tilde{A}_i(\tilde{P}_{i,k} + \tilde{A}_i^T\tilde{Q}_i^{-1}\tilde{A}_i)^{-1}\tilde{A}_i^T\tilde{Q}_i^{-1} = 0. \qquad (15)$$

C. Solving the ARE as an SDP

Even though the most efficient methods for solving AREs do not rely on SDPs, in the path for jointly solving the ARE and optimizing the trajectory, we first describe how to cast the solution of the ARE as an SDP. This will allow us to benefit from the efficient solvers available for SDPs and from their convexity properties in order to efficiently approach the persistent monitoring problem. Moving in this direction, we first introduce a relaxed version of (15), where equality is replaced by inequality, with the goal that, in the optimal solution of the optimization, the constraint will be tight and equality will hold:

$$\tilde{Q}_i^{-1} - \Pi_i + \tilde{H}_{i,k}^T\tilde{R}_i\tilde{H}_{i,k} \\ - \tilde{Q}_i^{-1}\tilde{A}_i(\Pi_i + \tilde{A}_i^T\tilde{Q}_i^{-1}\tilde{A}_i)^{-1}\tilde{A}_i^T\tilde{Q}_i^{-1} \succcurlyeq 0, \qquad (16)$$

where $\succeq 0$ denotes that the matrix is positive semi-definite and Π_i is a variable for which we want $\Pi_i = \tilde{P}_{i,k}$ as in (15) in the optimal solution of the optimization. We recall that whenever every target is visited, (15) will have a solution and hence the constraint 16 will be feasible. Using the Schur complement [10], this inequality can be written as:

$$\begin{bmatrix} \tilde{Q}_i^{-1} - \Pi_i + \tilde{H}_{i,k}^T \tilde{R}_i \tilde{H}_{i,k} & \tilde{Q}_i^{-1} \tilde{A}_i \\ \tilde{A}_i^T \tilde{Q}_i^{-1} & \Pi_i + \tilde{A}_i^T \tilde{Q}_i^{-1} \tilde{A}_i \end{bmatrix} \succeq 0. \quad (17)$$

We also define an upper bound Γ_i on the covariance matrix $\Gamma_i \succeq \Pi_i$, which in the optimal solution will coincide with the covariance matrix. Using Schur's complement, this upper bound can be expressed as:

$$\begin{bmatrix} \Gamma_i & I \\ I & \Pi_i \end{bmatrix} \succeq 0. \quad (18)$$

Now, we show that using an SDP, we can compute the exact solution of the information filter and that the relaxations we proposed will indeed be tight in an optimal solution. Moreover, we show that the cost function of the optimization is equal to the trace of the augmented steady state covariance matrix. Inspired by [10], where it is shown that the LQR Ricatti equation can be solved as an SDP, we give the following proposition:

Proposition 1: If the pair $(\tilde{A}_i, \tilde{H}_{i,k})$ is observable and Q_i and R_i are positive definite, then the optimal solution of the following SDP is such that $\Pi_i^* = \tilde{P}_{i,k}$ is a solution of the ARE (15) and $\Gamma_i^* = (\Pi_i^*)^{-1}$.

$$\min_{\Gamma_i, \Pi_i} \quad \mathrm{tr}(\Gamma_i)$$
$$\text{s.t.} \quad (17), \quad (18), \quad \Gamma_i, \ \Pi_i \succeq 0. \quad (19)$$

The proof of Prop. 1 is given in Appendix 1 of [11] and omitted here due to space reasons. In the optimal solution of the SDP, Γ_i^* is the augmented covariance matrix, i.e. $\Gamma_i^* = \tilde{P}_{i,k}^{-1}$. Therefore, minimizing $\mathrm{tr}(\Gamma_i)$ for all the targets is equivalent to minimizing $\tau^{-1} \sum_{i=1}^{\tau} \mathrm{tr}(\bar{\Sigma}_i(k))$, which is the optimization objective in the persistent monitoring problem, as expressed in (5). Thus, solving (19) gives us the squared estimation error of a single target, given an agent trajectory. Although Prop. 1 is not directly used to solve the Persistent Monitoring problem, it gives important insight into Prop. 2.

IV. OPTIMIZATION OF PERSISTENT MONITORING SCHEDULES

In this section, we give a procedure to jointly optimize the steady state uncertainty (5) and the trajectory of the agent. If we knew in advance when $\|s(k) - x_i\|$ was larger than r_i, then (3) would be linear with the squared distance between the agent and target i ($d_i^2(k)$) at every time for every i, and the problem would be an SDP. However, since we do not know whether or not $\|s(k) - x_i\|$ is larger than r_i, a set of SDPs needs to be solved in order to obtain the optimal trajectory with that period. Therefore, we propose in this section a two-step procedure. In the higher level, an algorithm produces sequences of targets to be visited by the agent and determines which of the modes of (3) is active in each time step. The lower level, on the other hand, assumes

a fixed mode in (3) given by the higher level algorithm and through the solution of an SDP produces a trajectory that minimizes the steady state uncertainty.

A. Lower Level Problem

Recalling (3), in order to simplify notation for the rest of this subsection, we define $G_i = H_{i,\max}^T R_i H_{i,\max}$ and its augmented version $\tilde{G}_i = diag(G_i, ..., G_i)$. Note that $H_{i,k}^T R_i H_{i,k} = \gamma_i(k) G_i$ and in the optimization $\gamma_i(k)$ will be treated as a decision variable and G_i as a constant. Therefore,

$$\tilde{H}_{i,k}^T \tilde{R}_i \tilde{H}_{i,k} = \underbrace{\begin{bmatrix} \gamma_i(k) I_{L_i \times L_i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \gamma_i(k+\tau) I_{L_i \times L_i} \end{bmatrix}}_{\tilde{\gamma}_{i,k}} \tilde{G}.$$
$$(20)$$

Moreover, we create optimization variables d_i^2 such that $d_i^2(k) \geq \|s(k) - x_i\|^2$. The underlying goal of creating this variable is that the constraint will be binding in an optimal solution, i.e., $d_i^2(k) = \|s(k) - x_i\|^2$, therefore we can compute $\gamma_i(k)$ using (3) once $d_i^2(k)$ is fixed.

Finally, we define logical variables $b_{i,k} \in \{0, 1\}$ (that will be fixed pre-defined variables in the optimization) as

$$b_{i,k} = \begin{cases} 0, & \|s(k) - x_i\| > r_i, \\ 1, & \|s(k) - x_i\| \leq r_i. \end{cases} \quad (21)$$

These logical variables represent whether or not the agent visits a given target i (i.e., the target within the agent's sensing range) at time step k of the cycle and thus define the mode in (3). With that in mind, we state Prop. 2.

Proposition 2: For fixed values of cycle length τ and logical variables $b_{i,k}$, the solution of the optimization (22), when it exists, minimizes the cost in (5), and the optimal trajectory $s^*(\cdot)$ satisfies dynamic constraints (2).

$$\min_{\Gamma_i, \Pi_i, s(\cdot)} \quad \frac{1}{\tau} \sum_{i=1}^{N} \mathrm{tr}(\Gamma_i)$$
$$\text{s.t.} \quad \begin{bmatrix} \tilde{Q}_i^{-1} - \Pi_i + \tilde{\gamma}_{i,k} \tilde{G}_i & \tilde{Q}_i^{-1} \tilde{A}_i \\ \tilde{A}_i^T \tilde{Q}_i^{-1} & \Pi_i + \tilde{A}_i^T \tilde{Q}_i^{-1} \tilde{A}_i \end{bmatrix} \succeq 0,$$
$$\begin{bmatrix} \Gamma_i & I \\ I & \Pi_i \end{bmatrix} \succeq 0,$$
$$\|s(k+1) - s(k)\|^2 \leq u_{\max}^2,$$
$$\|s(\tau) - s(1)\|^2 \leq u_{\max}^2,$$
$$\|s(k+1) - x_i\|^2 \leq d_i^2(k),$$
$$d_i^2(k) \geq r_i^2, \qquad \text{if } b_{i,k} = 0,$$
$$\gamma_i(k) = 0, \qquad \text{if } b_{i,k} = 0,$$
$$d_i^2(k) \leq r_i^2, \qquad \text{if } b_{i,k} = 1,$$
$$\gamma_i(k) = 1 - \frac{d_i^2(k)}{r_i^2}, \qquad \text{if } b_{i,k} = 1,$$
$$\forall i \in \{1, ..., N\}, \ \forall k \in \{1, ..., \tau\}. \quad (22)$$

A sketch of the proof of this proposition is given in [11]. Some brief insights on (22) are that the first three constraints

are used in the solution of the ARE, similar to Prop. 1. The following two constraints ensure that the agent movement is feasible according to (2) and that it is periodic. The next one is used to compute the distance between the agent and the target and the last four constraints compute γ_i based on the relative position of the agent and the target. We also note that constraints that involve the squared norm can be transformed into linear matrix inequalities using the Schur complement, therefore the optimization (22) can be cast as an SDP.

B. Higher Level Problem

Using the optimization problem (22), we have a procedure such that, for each cycle period τ, we can solve an exponentially growing ($2^{N \times \tau}$) number of SDPs, representing different variations of $b_{i,k}$, and obtain the optimal solution for that cycle period. This approach is very inefficient due to the exponential scalability. We thus propose a graph-based scheme that explores different combinations of τ and $b_{i,k}$ by exploring how targets are spatially distributed and evaluates each combination using the low level optimization (22).

As a motivation for the higher level algorithm we propose, consider the case where two targets are far enough apart that the agent seeing one target at a time instant cannot see the other one in the following time step due to the constraints in the dynamics. In a "blind" exploration of variables $b_{i,k}$, one could encode the possibility of the agent visiting these targets at consecutive time steps and such choice of $b_{i,k}$ renders an infeasible solution of (22). Moreover, trajectories that are dynamically feasible but have one target that is not visited may lead to unbounded cost and therefore we also do not want to explore them. The goal of the algorithm we introduce in this section is to evaluate only sets of τ and $b_{i,k}$ that can produce feasible trajectories according to the dynamics and the constraints as in (2) and also lead to bounded steady state uncertainty. The algorithm we present in this subsection is a "brute force" approach and we plan to explore more efficient exploration schemes in future work.

We abstract the targets as nodes in a graph $\mathcal{G}$. The goal is to find a sequence of nodes to be visited. The cost $\xi(i, h)$ of each edge (i, h) in the graph $\mathcal{G}$ is the minimum number of time steps necessary for the agent to transition between visiting these two targets, i. e.,

$$\xi(i, h) = \max \left(1, \left\lceil \frac{\|x_i - x_h\| - r_i - r_h}{u_{\max}} \right\rceil \right). \quad (23)$$

We point out that visiting a target means being located within its sensing range and not necessarily being exactly at its center. This is the reason why we subtract the radius in the numerator in (23). Also, note that if an agent is visiting a given target, it can visit the same target in the following time step. Therefore, the self-transition cost is such that $\xi_{i,i} = 1$, for any target. Given this structure, we can directly translate any sequence of visited nodes $\mathcal{S} = \{n_1, ..., n_F\}$ to the number of time steps in a cycle, τ, and to $b_{i,j}$, where

$$\tau(\mathcal{S}) = \xi(n_F, n_1) + \sum_{j=1}^{F-1} \xi(n_j, n_{j+1}). \quad (24)$$

We then propose Algorithm 1, which combines both low and high level stages and we name this *SDP-PM*. The intuition behind this algorithm is that, initially, all the cycles in which each target is visited for exactly one time step are added to a list and ordered according to the number of time steps in that particular cycle. Then, these cycles are explored in order. In the exploration, the cost of that particular cycle is evaluated and all the possibilities of visiting one new (or the same) targets in that cycle are added to the list $\mathcal{L}$. One thing to note is that the first cycle to be explored will always be the traveling salesman optimal solution and "simpler" cycles will always be explored first. A more detailed explanation of Alg. 1 that includes an example is given in [11].

Algorithm 1 SDP-PM

1: $OptCost \leftarrow \infty$
2: $OptCycle \leftarrow \emptyset$
3: $\mathcal{L} \leftarrow \emptyset$
4: **for** $\mathcal{S} \in permutation(1, ...N)$ **do**
5:      Add $(S, \tau(\mathcal{S}))$ to $\mathcal{L}$.
6: **for** $i \in (1, ..., N_{iter})$ **do**
7:      $\mathcal{S} \leftarrow removeFirst(\mathcal{L})$
8:      $cost = lowerLevelOptimization(\mathcal{S})$
9:      **if** $cost < OptCost$ **then**
10:          $OptCost \leftarrow cost$
11:          $OptCycle \leftarrow \mathcal{S}$
12:      **for** $newVertex \in (1, ..., N)$ **do**
13:          **for** $p \in (2, ...N_{\mathcal{S}})$ **do**
14:              $\mathcal{S}_{new} \leftarrow \{\mathcal{S}_{1:p-1}, newVertex, \mathcal{S}_{p:F_{\mathcal{S}}}\}$
15:              Add $(S_{new}, \tau(\mathcal{S}_{new}))$ to $\mathcal{L}$
16: **return** $OptCost, OptCycle$

V. SIMULATION RESULTS

We implemented the SDP-PM Alg. 1 with dynamics as in (1) with parameters

$$A_i = \text{diag}(1.1, 1.1), \quad Q_i = \text{diag}(0.1, 0.1), \quad (25)$$

and observation model as in (3) with parameters

$$H_{\max} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \quad R_i = \text{diag}(1, 1), \quad r_i = 0.6. \quad (26)$$

The agent maximum displacement in one time step is bounded by $u_{\max} = 0.33$.

To the best of the authors' knowledge, the only approaches proposed in the scientific literature similar enough to be used as a comparison are RRC and its variant RRC^* [1]. Note that, in its original form, RRC considers a different definition of $\gamma_i(k)$ than Eq. (3) and a cost different from (4). However, RRC can be trivially modified to match our definitions. We implemented and compared it to our approach (SDP-PM) in a simple environment and the results are shown in Fig. 1. Due to the random nature of RRCs, we ran it 5 independent times and we show its best, worst and average performances.
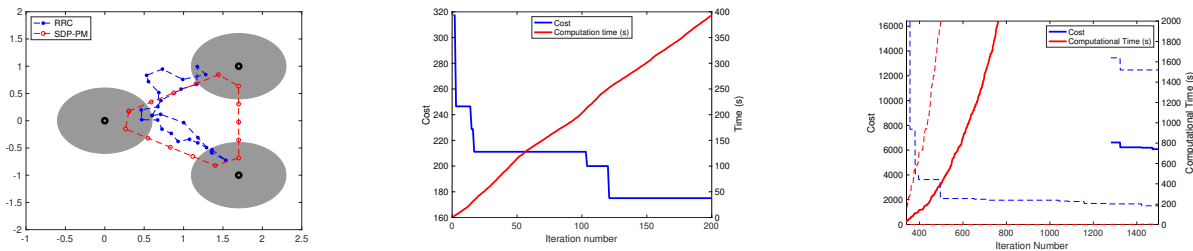
(a) Trajectories using SDP-PM and RRC. (b) Cost and computation time for SDP-PM. (c) Cost and computation time for RRC.

Fig. 1: Simulation results. (a) Comparison of the trajectories generated by the *RRC* and the *SDP-PM* approaches. The trajectory displayed for RRC is the one with lowest cost among 5 independent runs of the algorithm. The presented trajectory was obtained after 200 iterations of the SDP-PM algorithm and 1500 iterations of *RRC*. The grey area represents the positions for which the agent can sense a given target. (b) Cost and cumulative computation time as a function of the iteration number for *SDP-PM*. (c) Cost and cumulative computation time as a function of the iterations of *RRC*. The solid lines represent average among 5 runs and the dashed lines are the observed maximum and minimum of the cost and computational time. None of the 5 instances of *RRC* found a feasible solution before 345 iterations.

In Figs. 1b and 1c, each iteration of the *SDP-PM* consists in exploring one node (lines 6-14 of Alg. 1), while iterations of RRC consist of adding a node to the tree. When considering the computation time of RRC, in order to ensure a fair comparison, we only measured the time spent on solving the ARE, since it is the most computationally demanding part of RRC. The algebraic Riccati equations were solved using the MATLAB $idare$ function. In the case of the *SDP-PM*, the computation time in Fig. 1b corresponds to the time spent solving SDPs.

From Fig. 1a, one can see that the trajectory produced by *SDP-PM* travels between targets in a straight line, while *RRC* does not. Also, when the agent visits a target in the *SDP-PM* trajectory, it always moves as close as possible to the center of the target (given total time steps and speed limitations), which does not happen in *RRC*. The reason is that, for fixed τ and logical variables $b_{i,k}$, the trajectory generated by *SDP-PM* is optimal, while *RRC* only has an asymptotic probabilistic notion of optimality, with no deterministic guarantees for a finite number of iterations. Moreover, Figs. 1b and 1c show that, for reasonable computation times, *SDP-PM* produces much better solutions in terms of cost. The solution found at the first iteration of *SDP-PM* has the cost equal to 16.8% of the cost of the best (in terms of cost) of the 5 runs of *RRC* after 1500 iteration. Comparing our approach after 200 iterations and *RRC* after 1500, *SDP-PM* reports a cost of only 9% of the best solution of *RRC*. We also note that *SDP-PM* produced a solution with bounded cost in its first iteration, while *RRC* took between 345 and 1305 iterations to find its first feasible solution, i.e. where the target covariances are bounded. We refer the reader to [11] for simulations in more complex scenarios.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented an approach to compute persistent monitoring trajectories for systems that evolve in discrete time, with linear, stochastic dynamics. Given the optimal estimator (the Kalman Filter), we jointly optimized the cost and the position of the agents, in a way that minimizes the infinite horizon cost. The simulation results have shown that our approach efficiently solves small sized problems and that it significantly improves the state of the art, both in terms of computational cost and performance.

For future work, we intend to extend the proposed technique to multi-agent systems and plan to analyze ways to overcome the numerical issues with the SDP that arise when the size of the problem grows. Finally, we also want to improve the higher level search algorithm, enhancing the way we explore the candidate optimal cycles, possibly by greedily exploring from an initial candidate schedule.

REFERENCES

[1] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[2] M. Ostertag, N. Atanasov, and T. Rosing, "Robust Velocity Control for Minimum Steady State Uncertainty in Persistent Monitoring Applications," in *American Control Conference*, 2019, pp. 2501–2508.

[3] X. Lan and M. Schwager, "A Variational Approach to Trajectory Planning for Persistent Monitoring of Spatiotemporal Fields," in *American Control Conference*, 2014, pp. 5627–5632.

[4] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Multi-Agent Infinite Horizon Persistent Monitoring of Targets with Uncertain States in Multi-Dimensional Environments," in *IFAC World Congress (to appear)*, 2020.

[5] S. Thrun, "Probabilistic Robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[6] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Optimal Periodic Multi-Agent Persistent Monitoring of a Finite Set of Targets with Uncertain States," in *American Control Conference (to appear)*, 2020.

[7] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the Optimal Solutions of the Infinite-horizon Linear Sensor Scheduling Problem," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2825–2830, 2014.

[8] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[9] K. Fujimoto, Y. Oji, and K. Hamamoto, "On Periodic Kalman Filters and Multi-rate Estimation," in *IEEE Conference on Control Applications*, 2016, pp. 934–939.

[10] V. Balakrishnan and L. Vandenberghe, "Connections Between Duality in Control Theory and Convex Optimization," in *American Control Conference*, vol. 6, 1995, pp. 4030–4034.

[11] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "A Semidefinite Programming Approach to Discrete-time Infinite Horizon Persistent Monitoring," in *ArXiV ID 2104.00166*, 2020.