Multi-robot persistent environmental monitoring based on constraint-driven execution of learned robot tasks

Gennaro Notomista, Claudio Pacchierotti, Paolo Robuffo Giordano

Multi-Robot Persistent Environmental Monitoring Based on Constraint-Driven Execution of Learned Robot Tasks

Gennaro Notomista, Claudio Pacchierotti, and Paolo Robuffo Giordano

***Abstract*— This paper considers a multi-robot team tasked with monitoring an environmental field of interest over long time horizons. The approach is based on a control-theoretic measure of the information collected by the robots, namely a norm of the constructability Gramian. This measure is leveraged in order to learn a distributed multi-robot control policy using the reinforcement learning paradigm. The learned policy is then combined with energy constraints using the constraint-driven control framework in order to achieve persistent environmental monitoring. The proposed approach is tested in a simulated multi-robot persistent environmental monitoring scenario where a team of robots with limited availability of energy is to be controlled in a coordinated fashion in order to estimate the concentration of a gas diffusing in the environment.**

I. Introduction

Robotic systems are increasingly common in environmental monitoring applications owing to their suitability to collect spatially and temporally disperse data [1]. In particular, multi-robot systems naturally lend themselves to such applications thanks to the large footprint they can achieve despite the simplicity of each robotic unit. Whenever environmental data need to be collected over long time horizons, hardware solutions—such as larger energy storage devices and solar-powered robots—become less effective. In these challenging situations, energy management and control are paramount [2].

The constraint-driven control paradigm, introduced in [3], has proven to be suitable for long-duration robot autonomy for two main reasons: (i) Its optimization-based formulation aims at minimizing the control effort spent by the robots while executing a given task, and (ii) it provides a framework to constrain the execution of robotic tasks by the energy availability of the robots. Applications of the constraint-driven control paradigm range from the persistification of robotic tasks [4] to coordinated-control of robot teams [5], to energy-optimal multi-agent motion planning [6] and energy-aware multi-robot task allocation [7], [8].

The approach presented in this paper is based on the one proposed in [9], where a persistent environmental monitoring strategy is proposed, which consists in optimizing the robot trajectory in order to maximize the information gathered while traversing it. This approach has two advantages: (i) It possesses theoretical guarantees in terms of information collected to reconstruct an environmental field of interest, and (ii) the environment monitoring task lends itself to be embedded in a constraint-driven control framework. The latter, given the amenability of the constraint-driven control framework for long-duration robot autonomy, allows for the implementation of long-term environmental monitoring.

The environmental monitoring approach in [9] based on active sensing, however, assumes the robotic system to be differentially flat in order to compute the control input required to track the optimized state trajectory. In general, this is not the case when considering multi-robot systems controlled in a coordinated fashion [10]. Moreover, compared to the single-robot case considered in [9], where the measurements collected by a robot are used by itself to plan the most informative trajectory, in multi-robot scenarios it is not clear how collected measurements should be utilized by the team of robots to estimate an environmental field of interest in a distributed fashion.

In this paper, leveraging the insights of the control-theoretic active sensing approach, we propose to extend the approach proposed in [9] by considering a multi-robot setting. This is achieved by learning a coordinated-control policy for multi-robot environmental monitoring tasks using the reinforcement learning (RL) paradigm. The learned policy in then combined with energy constraints to achieve multi-robot persistent environmental monitoring.

A. Related Work

The problem of multi-robot environmental monitoring typically consists in reconstructing physical processes evolving, in general, both in space and in time, using a team of autonomous robots. This topic has been extensively studied in the recent years [11], [12], [13], [14], [15], [1], [16].

In [11], most-informative paths are planned for multiple autonomous robots in order to achieve an efficient collection of environmental data. The authors propose an approach which shares the informative multi-robot path planning feature with the control strategy proposed in this paper. Differently from the constraint-driven formulation of persistent environmental monitoring that we develop in this paper, however, limited availability of energy is not explicitly taken into account, as it is assumed that the robots always have energy resources at their disposal.

Other approaches consist in minimizing energetically-expensive operations, such as accelerating and stopping mobile robotic platforms, as done in [14]. Compared to the persistent environmental monitoring approach proposed in this paper, these approaches are focused on optimizing energy consumption rather than constraining the robot motion required to collect environmental data by the energy stored

This works was sponsored by the ANR project ANR-20-CHIA-0017 "MULTISHARED".

The authors are with CNRS, Univ Rennes, Inria, IRISA, Rennes, France {gennaro.notomista,claudio.pacchierotti,prg}@irisa.fr

in the batteries of the robots. The latter is the main point that characterizes the control strategy we propose here.

B. Contributions and Paper Organization

The contributions of this paper are twofold and can be summarized as follows:

(i) We extend the persistent environmental monitoring approach developed in [9] to the multi-robot case by learning a coordinated-control policy using the RL paradigm which leverages the control-theoretic measures of information collected in the environment

(ii) We show how to execute the learned multi-robot policy by means of the RL paradigm in a constraint-driven control framework, which allows us to combine the environmental monitoring task with energy control in order to achieve persistent environmental monitoring

The remainder of the paper is organized as follows. Section II briefly introduces the constraint-driven task execution framework and the formulation of environmental monitoring as an active sensing problem. Section III is devoted to the design of the multi-robot environmental monitoring control policy using RL. Section IV provides the theoretical connection between policies learned using the RL paradigm and the execution of the corresponding tasks using the constraint-driven control framework, and it presents the optimization problem designed to achieve persistent environmental monitoring. Section V presents and discusses the simulation results and Section VI concludes the paper.

II. Background

In this paper, we employ the constraint-driven robot control paradigm in order to let a multi-robot system perform environmental monitoring over long-time horizons. This section introduces the two main components required to achieve this goal, namely the constraint-driven task execution framework and the environmental monitoring formulation based on active sensing.

A. Constraint-Driven Task Execution

The general form of the constraint-driven control paradigm to execute a task can be stated as follows [3]:

$$\begin{aligned} &\underset{u}{\text{minimize}} \ \|u\|^2 \\ &\text{subject to } c_{\text{task}}(x,u) \leq 0, \end{aligned} \tag{1}$$

where x and u are state and control input of the robotic system and c_{task} is a function encoding the task to execute. In this paper, c_{task} will be used to encode the multi-robot environmental monitoring task. Notice how the task is encoded as a constraint of an optimization program whose objective is to minimize the control effort u: This is the first reason that renders this formulation amenable for long-duration autonomy. The second reason is the fact that the optimization-based nature of the formulation allows us to enforce energy-control as an additional constraint, as follows:

$$\begin{aligned} &\underset{u,\delta}{\text{minimize}} \ \|u\|^2 + \kappa\delta^2 \\ &\text{subject to } c_{\text{task}}(x,u) \leq \delta \\ &\qquad\qquad c_{\text{energy}}(x,u) \leq 0. \end{aligned} \tag{2}$$

Here, c_{energy} is used to constrain the task execution by the current availability of energy of the robots, and it can be defined, for instance, in order to prevent the energy stored in the robot batteries from going below a desired minimum threshold. The slack variable δ is used to prioritize the energy constraint over the task execution in order to effectively realize a persistent execution of the robotic task. $\kappa > 0$ is an optimization parameter.

To make the exposition more concrete, consider a robotic system with control affine dynamics

$$\dot{x} = f_0(x) + f_1(x)u, \tag{3}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ denote state and control input, respectively, and $f_0 \colon \mathbb{R}^n \to \mathbb{R}^n$ and $f_1 \colon \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are Lipschitz continuous vector fields. The task to be executed is encoded by a continuously differentiable, positive definite cost function $V \colon \mathcal{X} \to \mathbb{R}_+$. High values of $V(x)$ correspond to the task being far from being completed, while $V(x) = 0$ corresponds to the task being accomplished. With this notation, the optimization program (1) can be expressed as follows:

$$\begin{aligned} &\underset{u}{\text{minimize}} \ \|u\|^2 \\ &\text{subject to } L_{f_0}V(x) + L_{f_1}V(x)u + \gamma_V(V(x)) \leq 0, \end{aligned} \tag{4}$$

where $L_{f_0}V(x)$ and $L_{f_1}V(x)$ are the Lie derivative of V along the vector fields f_0 and f_1, respectively. The function $\gamma_V \colon \mathbb{R} \to \mathbb{R}$ is a Lipschitz continuous extended class $\mathcal{K}$ function—i.e. a Lipschitz continuous, monotonically increasing function, with $\gamma_V(0) = 0$. The constraint $L_{f_0}V(x) + L_{f_1}V(x)u + \gamma_V(V(x)) \leq 0$ can be equivalently written as $\dot{V}(x,u) + \gamma_V(V(x)) \leq 0$ which ensures that $V(x(t)) \to 0$ as $t \to \infty$.

The following section will show how (2) can be used to execute a (single-robot) persistent environmental monitoring task. This is achieved using the control-theoretic notion of constructability Gramian, $\mathcal{G}_c \in \mathbb{R}^n \times \mathbb{R}^n$, whose trace can be used as a metric to quantify the amount of information collected during an estimation process. See also Remark 5 in [9] for more detailed theoretical justifications of this choice.

B. Persistent Environmental Monitoring as Active Sensing

The compound model adopted in this paper for the robot, the energy, and the environmental field dynamics is the following:

$$\begin{cases} \dot{x} = f_0(x) + f_1(x)u \\ \dot{e} = -\dfrac{\eta}{C}\beta\left(u_{\max}\right) \\ \dot{\theta} = f_\theta(\theta, t) \\ y = f_y(x, \theta), \end{cases} \tag{5}$$

where $u_{\max}$ is the maximum norm of the control input u, $e \in \mathbb{R}$ represents the energy stored in the robot battery. The

dynamics of e depend on the physical parameters η and C, which correspond to the efficiency and the capacity of the robot battery, and on β, a monotonically increasing function that models the dependency of the current flowing out the battery on the control input u. See [17] for more details and discussion on the energy model.

We consider environments where there exists a scalar field representing a quantity of interest whose distribution is to be estimated by the robots. The environmental field is parameterized by the vector of parameters $\theta \in \mathbb{R}^{n_\theta}$ whose dynamics may depend on θ itself and on time t. The components of θ can represent, for instance, the location of a source or the rate of expansion of a chemical substance in the environment. This environmental model is general as it encompasses environmental fields whose dynamics evolve according to ordinary and partial differential equations alike. Finally, the measurement model is represented by the function f_y whose value—the measurement—depends both on x and θ.

The approach for environmental monitoring based on active sensing is proposed in [9] and consists in optimizing a state trajectory in order to maximize the information collected by the robot while traversing it. This information is measured by the trace of the inverse of the so-called constructability Gramian. As a result, the cost function V encoding the environmental monitoring task can be chosen to be

$$V(x(t)) = \operatorname{trace}\left(\mathcal{G}_c^{-1}(x(t))\right)^2, \tag{6}$$

and

$$\mathcal{G}_c(x(t)) = \int_{-\infty}^{t} \Phi(\tau,t)^T C(\tau)^T W(\tau) C(\tau) \Phi(\tau,t) \mathrm{d}\tau, \tag{7}$$

where $\Phi(\tau,t) \in \mathbb{R}^{n\times n}$ is the transition matrix of the dynamical system (3), $C(\tau) = \frac{\partial f_y(x(\tau))}{\partial x}$, and W is a weight matrix. The lower the trace of the inverse of the constructability Gramian, the more information is collected by the robot until time t, and so the closer the robot is to accomplishing the environmental monitoring task. In [18] and [9], an equivalent approach is considered, consisting in maximizing the trace of the constructability Gramian.

The constraint $c_{\text{task}}(x,u) \leq \delta$ in (2) then becomes $\dot{V}(x,u) + \gamma_V(V(x)) \leq \delta$, where $V(x)$ is given by (6). To conclude this section on persistent environmental monitoring based on active sensing, we introduce the energy constraint c_{energy} similar to the one proposed in [9], which can be enforced in (4), with V given by (6), in order to achieve persistent environmental monitoring. Given the model (5) for the dynamics of x and e, $c_{\text{energy}}(x,u) \leq 0$ turns into $-\dot{h}(x,e,u) - \gamma_h(h(x,e)) \leq 0$, where h is the following control barrier function:

$$h(x,e) = e - e_{\min} - \alpha_c\left(\|p(x) - p_c\|\right), \tag{8}$$

and $\gamma_h \colon \mathbb{R} \to \mathbb{R}$ is a Lipschitz continuous extended class $\mathcal{K}$ function. Control barrier functions [19] are typically used to enforce forward-invariance constraints on subsets of the state space of dynamical systems. For the purposes of energy control, we employ control barrier functions to prevent the energy e from going below the minimum threshold $e_{\min}$. In the definition of h in (8), $p(x)$ denotes the robot position in space (e.g., $p(x) \in \mathbb{R}^2$ for planar robots, $p(x) \in \mathbb{R}^3$ for aerial robots), and p_c is the spatial location of a charging station, i.e. a place that the robot can reach in order to recharge its battery. α_c is a monotonically increasing function, so that the quantity $\alpha_c\left(\|p(x) - p_c\|\right)$ represents the energy required to reach the charging station located at p_c from location $p(x)$.

Remark 1 (Active sensing in multi-robot settings)**.** *The constructability Gramian defined in* (7) *depends on the state trajectory until current time t. In order to relate the state trajectory to the input required by the robot to track it, in [18], the authors assume that the system is differentially flat. While this is the case for many mobile robotic platforms (both ground and aerial), coordinated multi-robot systems do not always exhibit this property. Additionally, in multi-robot settings, it is not clear how to combine the measurements collected by all the robots using the active sensing approach developed in [9] for a single robot.*

To overcome the problem highlighted in Remark 1, in the following section, we propose an approach to learn multi-robot environmental monitoring policies, which leverages the RL paradigm. This approach is based on the control-theoretic insights of the constructability Gramian used in the active sensing formulation.

III. Multi-Robot Persistent Environmental Monitoring

Let us consider N robots, modeled by the dynamics in (5) where we will use subscript i to refer to quantities related to the i-th robot of the team, i.e.:

$$\begin{cases} \dot{x}_i = f_0(x_i) + f_1(x_i)u_i \\ \dot{e}_i = -\dfrac{\eta}{C}\beta\left(u_{\max}\right) \\ \dot{\theta} = f_\theta(\theta,t) \\ y_i = f_y(x_i,\theta). \end{cases} \tag{9}$$

In this paper, we consider homogeneous robot teams, where all robots share the same dynamics (i.e. η, C, β, f_θ, and f_y are the same for all robots). A straightforward extension can be made to encompass heterogeneous robot teams as well.

Compared to individual robots, multi-robot systems can be controlled in a coordinated fashion with the objective of realizing ensemble behaviors, such as spreading over an environment or maintaining connectivity among the team. In order to enforce such a coordination between the robots, the expression of f_0 and f_1 can be chosen to be as follows:

$$\dot{x}_i = \frac{\partial \mathcal{E}}{\partial x_i}^T(x) + u_i, \tag{10}$$

where $\mathcal{E} \colon \mathbb{R}^{Nn} \to \mathbb{R}$ is a performance cost which can be used to encode a wide range of multi-robot coordinated behaviors, such as consensus, formation control, and coverage control [10], and $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^{Nn}$ denotes the compound state of the multi-robot system. As pointed out in Remark 1,

these coordinated behaviors—which are extensively used to control robotic systems comprised of multiple interconnected robotic units—might prevent the system from being differentially flat. Thus, the active-sensing-based approach proposed in [9] for the single-robot case cannot be extended to the multi-robot case in a straightforward fashion.

Therefore, in this paper, we take a different approach and *learn* the multi-robot control policy required to estimate the environmental field of interest. Using the RL paradigm, starting from the definition of a reward assigned to each state of the system (9), denoted by $g(x_i, u_i)$, an approximation of the so-called value function can be evaluated using an RL algorithm [20]. The learned approximate value function—which will be denoted in the following by $\tilde{J}^\star$—has the same meaning of the cost function V introduced in Section II-A, i.e. higher values correspond to the learned task being far from being completed, while the task is accomplished when $\tilde{J}^\star$ is minimized.

The reward function used in this paper for the multi-robot environmental monitoring task is defined as follows:

$$g(x_i, u_i) = \operatorname{trace}\left(\mathcal{G}_c(x_i(t))\right)^2. \tag{11}$$

Based on the discussion in Section II-B, high rewards correspond to large amounts of information collected until time t. Using an RL algorithm, the approximate value function $\tilde{J}^\star$ can be obtained, and, in the next section, we will show the connection between RL (approximate dynamic programming) and constraint-driven control. This will allow us to leverage the learned value function in the constraint-driven task execution framework in order to combine the environmental monitoring objective with the energy constraint.

IV. From Dynamic Programming to Constraint-Driven Control

Using the system model (9) to describe the motion of each robot of the team, together with the reward function (11), an approximation for the value function $\tilde{J}^\star$, can be obtained using an RL algorithm. Owing to the definition of the reward, and as confirmed by the results in the next section, this value function effectively encodes the desired environmental monitoring task.

Once $\tilde{J}^\star$ has been obtained, we are interested in executing the optimal policy corresponding to this value function. In this section, we show how this can be done within the constraint-driven control framework, which allows us to enforce additional constraints—as, for instance, energy constraints—on top of the optimal policy to render the multi-robot environmental monitoring task persistent.

With this goal in mind, let us start by considering a system with known discrete-time dynamics

$$x_{k+1} = \hat{f}(x_k, u_k),$$

where x_k denotes the state, $u_k \in \mathcal{U}_k(x_k)$ the input, and the input set $\mathcal{U}_k(x_k)$ may depend in general on the time k and the state x_k. The value iteration algorithm to solve a deterministic dynamic programming problem with no terminal cost can be stated as follows [20]:

$$J_{k+1}(x_k) = \min_{u_k \in \mathcal{U}_k(x_k)} \Big\{ g_k(x_k, u_k) + J_k(\hat{f}(x_k, u_k)) \Big\}, \tag{12}$$

with $J_0(x_0) = 0$, where x_0 is the initial state, and $g_k(x_k, u_k)$ is the cost incurred at time k. J corresponds to an accumulated cost, and the total cost accumulated along the system trajectory is given by

$$J(x_0) = \lim_{N \to \infty} \sum_{k=0}^{N-1} \alpha^k g_k(x_k, u_k). \tag{13}$$

In this paper, we will consider $\alpha = 1$ and we will assume there exists a cost-free termination state.By Proposition 4.2.1 in [20] the value iteration algorithm (12) converges to the value function $J^\star$ that satisfies the following equation:

$$J^\star(x) = \min_{u \in \mathcal{U}(x)} \Big\{ g(x, u) + J^\star(\hat{f}(x, u)) \Big\}. \tag{14}$$

Adopting a so-called approximation scheme in value space, $J^\star$ can be replaced by a parametric approximation $\tilde{J}^\star$ by solving the following approximate dynamic programming algorithm:

$$\tilde{J}_{k+1}(x_k) = \min_{u_k \in \mathcal{U}_k(x_k)} \Big\{ g_k(x_k, u_k) + \tilde{J}_k(\hat{f}(x_k, u_k)) \Big\}.$$

In these settings, RL algorithms can be leveraged to find parametric approximations, $\tilde{J}^\star$, of the value function using neural networks—such algorithms are commonly referred to with the name of deep RL. This is the paradigm considered in this paper in order to approximate the value function encoding the multi-robot environmental monitoring task.

The bridge between dynamic programming and constraint-driven control is optimal control. In fact, the cost in (13) is typically considered in optimal control problems, recalled, in the following, for the continuous time control affine system (3):

$$\begin{aligned} &\underset{u(\cdot)}{\text{minimize}} \int_0^\infty \big(q(x(t)) + u(t)^T u(t)\big)\, \mathrm{d}t \\ &\text{subject to } \dot{x} = f_0(x) + f_1(x)u. \end{aligned} \tag{15}$$

Comparing (15) with (13), we recognize that the instantaneous cost $g(x, u)$ in (13) in the context of the optimal control problem (15) corresponds to $q(x) + u^T u$, where $q\colon \mathcal{X} \to \mathbb{R}$ is a continuously differentiable and positive definite function.

A dynamic programming argument on (15) leads to the following Hamilton-Jacobi-Bellman equation:

$$L_{f_0} J^\star(x) - \frac{1}{4} L_{f_1} J^\star(x) \left(L_{f_1} J^\star(x)\right)^T + q(x) = 0,$$

where J^* is the value function—similar to (14) for continuous-time problems—representing the minimum cost-to-go from state x, defined as

$$J^\star(x) = \min_{u(\cdot)} \int_t^\infty \big(q(x(\tau)) + u(\tau)^T u(\tau)\big)\, \mathrm{d}\tau. \tag{16}$$

The optimal policy corresponding to the value function (16) can be evaluated as follows [21]:

$$u^\star = -\frac{1}{2}\left(L_{f_1}J^\star(x)\right)^T. \tag{17}$$

In order to show how the optimal policy $u^\star$ in (17) can be obtained using an optimization-based formulation, we now recall the concept of control Lyapunov functions.

Definition 1 (Control Lyapunov function [22])**.** *A continuously differentiable, positive definite function $V\colon \mathbb{R}^n \to \mathbb{R}$ is a control Lyapunov function (CLF) for the system* (3) *if, for all $x \neq 0$*

$$\inf_u \left\{ L_{f_0}V(x) + L_{f_1}V(x)u \right\} < 0. \tag{18}$$

To select a control input u which satisfies the inequality (18), an expression—known as the Sontag's formula [23]—can be employed. With the aim of encoding the optimal control input $u^\star$ by means of a CLF, we will consider the following modified Sontag's formula originally proposed in [24]:

$$u(x) = \begin{cases} -v(x)\left(L_{f_1}V(x)\right)^T & \text{if } L_{f_1}V(x) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where $v(x) = \frac{L_{f_0}V(x) + \sqrt{\left(L_{f_0}V(x)\right)^2 + q(x)L_{f_1}V(x)\left(L_{f_1}V(x)\right)^T}}{L_{f_1}V(x)\left(L_{f_1}V(x)\right)^T}$.

The modified Sontag's formula (19) is equivalent to the solution of the optimal control problem (15) if the following relation between the CLF V and the value function $J^\star$ holds [25]:

$$\frac{\partial J^\star}{\partial x} = \lambda(x)\frac{\partial V}{\partial x}, \tag{20}$$

where $\lambda(x) = 2v(x)\left(L_{f_1}V(x)\right)^T$. The relation in (20) corresponds to the fact that the level sets of the CLF V and those of the value function $J^\star$ are parallel.

The last step towards the constrained-optimization-based approach to generate optimal control policies is to recognize the fact that, owing to its inverse optimality property, the modified Sontag's formula (19) can be obtained using the following constrained-optimization formulation, also known as the pointwise min-norm controller [24]:

$$\begin{aligned} &\underset{u}{\text{minimize}} \ \|u\|^2 \\ &\text{subject to } L_{f_0}V(x) + L_{f_1}V(x)u + \sigma(x) \leq 0, \end{aligned} \tag{21}$$

where $\sigma(x) = \sqrt{\left(L_{f_0}V(x)\right)^2 + q(x)L_{f_1}V(x)\left(L_{f_1}V(x)\right)^T}$.

This formulation already resembles the one introduced in (4) in Section II-A, and in the following we provide a formulation which strengthens the connection with approximate dynamic programming. When $V = \tilde{J}^\star$, the min-norm controller solution of (21) is the optimal policy which would be learned using an RL algorithm: this is what allows us to bridge the gap between constraint-driven control and RL.

Following the formulation in (4), the constraint-driven execution of a task encoded by the approximate value function $\tilde{J}^\star$ learned using RL can be implemented executing the control input solution of the following optimization program:

$$\begin{aligned} &\underset{u}{\text{minimize}} \ \|u\|^2 \\ &\text{subject to } \frac{1}{\lambda(x)}\left(L_{f_0}\tilde{J}^\star(x) + L_{f_1}\tilde{J}^\star(x)u\right) + \sigma(x) \leq 0. \end{aligned} \tag{22}$$

Remark 2 (Computation of task constraint using deep RL)**.** *The Lie derivatives $L_{f_0}\tilde{J}^\star(x)$ and $L_{f_1}\tilde{J}^\star(x)$ contain the gradients $\frac{\partial J^\star}{\partial x}$. In deep RL settings, when $\tilde{J}^\star(x)$ is approximated using neural networks, these gradients can be efficiently computed using back propagation.*

To summarize, using an RL algorithm, one can get the approximate value function $\tilde{J}^\star$, which can be then plugged in (22). The robot controlled using the solution of (22) executes the task encoded by the value function $\tilde{J}^\star$ in an optimal fashion.

We conclude this section by stating the main convex quadratic program which is used in the next section to synthesize the controller to let each robot of the team execute persistent environmental monitoring:

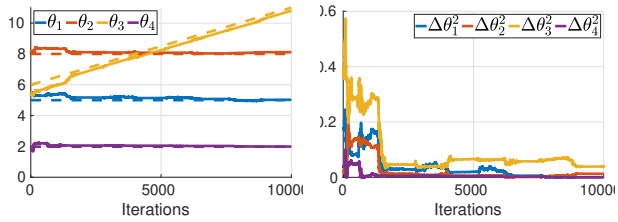
$$\begin{aligned} &\underset{u_i,\delta_i}{\text{minimize}} \ \|u_i\|^2 + \kappa\delta_i^2 \\ &\text{subject to } \frac{1}{\lambda(x)}\left(L_{f_0}\tilde{J}^\star(x) + L_{f_1}\tilde{J}^\star(x)u_i\right) + \sigma(x) \leq \delta_i \\ &\qquad - \dot{h}(x_i, e_i, u_i) - \gamma_h(h(x_i, e_i)) \leq 0, \end{aligned} \tag{23}$$

where x is the compound state of the multi-robot system, and δ_i, as discussed in Section II-A, prioritizes the energy constraint of robot i over the execution of the task encoded by the approximate value function $\tilde{J}^\star$.

V. Simulation Results

The persistent environmental monitoring strategy formulated in Sections III and IV has been tested in simulation on a team of 6 ground mobile robots deployed in a planar environment. The state x_i of robot i denotes its position in the plane, i.e. $p(x_i) = x_i \in \mathbb{R}^2$. The environmental field of interest consists of the concentration of a gas that spreads from an unknown source with the diffusive dynamics $f_y\colon (x_i, \theta) \mapsto \theta_4 e^{-\frac{\left\|p(x_i) - [\theta_1, \theta_2]^T\right\|^2}{\theta_3}}$ [26], where $[\theta_1, \theta_2]^T$ represents the (unknown) source location, θ_3 and θ_4 determine the (unknown) spread of the gas concentration, and $p(x_i)$ is the position of robot i. The dynamics of θ are given by $f_\theta\colon (\theta, t) \mapsto \begin{bmatrix} 0 & 0 & K_1 & -K_2 t^{-\frac{3}{2}} \end{bmatrix}^T$, with $K_1 = 0.005$ and $K_2 = 0.01$. Measurement noise parameters are the same as in [9].

Dedicated charging stations for each robot have been arranged in an hexagonal formation around the origin of the reference frame defined in the environment (see colored circles labeled as $p_{c,1}, \ldots, p_{c,6}$ in Fig. 3a on page 6). The parameters of the energy model in (9) have been set to: $\beta(s) = 0.005s$, $\eta = 1$, and $C = 1$. Following the derivation in [17], the energy required by robot i to reach the charging station located at $p_{c,i}$, used in the definition of the control



(a) (b)

Fig. 1. Figure 1a: Environmental field parameters estimated by robot 1 during the course of the simulated experiment (dashed line denote the ground truth values). Figure 1b: Squared estimation errors, $\Delta\theta_i^2$, with respect to ground truth values of the environmental field parameters.

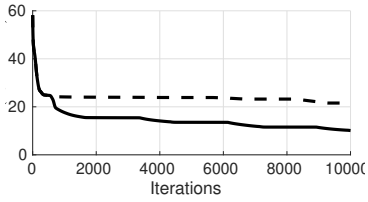
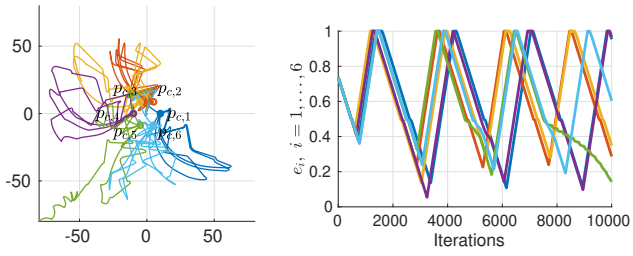


Fig. 2. Information collected by robot 1 during the estimation task, measured by the trace of the inverse of the constructability Gramian (smaller values correspond to more information collected). The dashed line corresponds to the case where robot 1 optimizes its trajectory without coordinating with the robot team, using the approach in [9].



(a) (b)

Fig. 3. Trajectories (Fig. 3a) and energy values (Fig. 3b) of the robots, recorded during the course of the experiment (each color corresponds to a different robot). In Fig. 3a, charging stations are depicted as colored circles and are labeled as $p_{c,1}, \ldots, p_{c,6}$. The robots visit their dedicated charging station when their batteries are depleting, driven by the second constraint in (23).

barrier function (8), is evaluated as $\alpha_c(\|p(x_i) - p_{c,i}\|) = \log(\|p(x_i) - p_{c,i}\|) - \log(d_{\max})$, where $d_{\max}$ is the maximum distance from the charging station located at $p_{c,i}$ at which robot i is able to recharge its battery. In the case of wireless charging technologies, $d_{\max}$ can be interpreted as the footprint of the wireless charging station. In the simulations, $d_{\max} = 0.25$ m, and $e_{\min} = 0$—i.e. the batteries should never be fully depleted. Additionally, to simulate the battery recharging process, $\dot{e} = 0.001$ s^{-1} when the robot is at a position within $d_{\max}$ from the charging station.

The coordination among the robots has been ensured by setting $\mathcal{E}(x) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|p(x_i) - p(x_j)\|^2$ in (10), where

$$\mathcal{N}_i = \begin{cases} \{2\} & i = 1 \\ \{i-1, i+1\} & 2 \leq i \leq 5 \\ \{5\} & i = 6. \end{cases}$$

This performance cost function $\mathcal{E}$ results in what is known as position consensus on a line graph [27], and can be employed to keep the members of the multi-robot system close to each other in order to facilitate information exchange. In fact, at each time instant t, we let the robots that are closer than 10 m from each other exchange the measurements they collected until time t.

In the optimization program (23), $\tilde{J}^\star$ is learned in an on-policy fashion using the Spinning Up vanilla policy gradient algorithm (`https://spinningup.openai.com/en/latest/`), $\kappa = 1$, and $\gamma_h(s) = s$.

Figure 1 shows the environmental field parameters as estimated by robot 1. As can be seen, the estimated values (solid lines) converge to the ground truth values (dashed lines), and after 2000 iterations reach a steady-state low estimation error. Another measure of the effectiveness of the environmental estimation process is the trace of the inverse of the constructability Gramian, which is inversely proportional to the amount of information collected by the robots. This metric is reported in Fig. 2: Here, the solid line corresponds to the value computed by robot 1, in the multi-robot scenario, during the course of the experiment. As a reference, the dashed line represents the trajectory of $\text{trace}\left(\mathcal{G}_c^{-1}(x_1(t))\right)$ computed using the single-robot estimation strategy proposed in [9], i.e. assuming each robot optimizes its own trajectory without accounting for the presence of the other robots of the team.

Finally, Fig. 3a shows the trajectories of the robots in the environment, recorded during the course of the simulated experiment. As can be seen, the robots routinely visit their dedicated charging station—depicted as a circle labeled by $p_{c,i}$ and plotted in the same color as the trajectory of the corresponding robot—when in need of energy, thanks to the effect of the second constraint in (23). As a result, the batteries of the robots never deplete and the energy is always kept above 0, as shown in Fig. 3b. This demonstrates the effectiveness multi-robot persistent environmental monitoring task learned and executed using the constraint-driven control framework. The next two remarks conclude this section by highlighting two additional amenable properties of the approach presented in this paper.

VI. Conclusion

In this paper, we presented a multi-robot persistent environmental monitoring strategy, suitable to control a team of robots deployed in an environment to estimate an unknown environmental field of interest. The robots are trained to learn a coordinated estimation strategy using the reinforcement learning paradigm. The learned policy is then combined with energy constraints in order to keep the energy stored in the batteries of the robots always above a minimum threshold during the execution of the estimation task. This is realized thanks to the constraint-driven task execution framework. The approach has been tested in simulation on a team of 6 mobile robots tasked with estimating the concentration of a gas diffusing in an environment from an unknown source.

REFERENCES

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[2] B. M. Sadler, "Fundamentals of energy-constrained sensor network systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 17–35, 2005.

[3] G. Notomista and M. Egerstedt, "Constraint-driven coordinated control of multi-robot systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 1990–1996.

[4] ——, "Persistification of robotic tasks," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 756–767, 2020.

[5] K. M. Cabral, S. N. Givigi, and P. T. Jardine, "Autonomous assembly of structures using pinning control and formation algorithms," in *2020 IEEE International Systems Conference (SysCon)*. IEEE, 2020, pp. 1–7.

[6] L. E. Beaver, M. Dorothy, C. Kroninger, and A. A. Malikopoulos, "Energy-optimal motion planning for agents: Barycentric motion and collision avoidance constraints," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 1040–1045.

[7] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An optimal task allocation strategy for heterogeneous multi-robot systems," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2071–2076.

[8] G. Notomista, S. Mayya, Y. Emam, C. Kroninger, A. Bohannon, S. Hutchinson, and M. Egerstedt, "A resilient and energy-aware task allocation framework for heterogeneous multirobot systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 159–179, 2022.

[9] 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.

[10] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[11] K.-C. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, "Multi-robot informative and adaptive planning for persistent environmental monitoring," in *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 285–298.

[12] J. J. Roldán, P. Garcia-Aunon, M. Garzón, J. De León, J. Del Cerro, and A. Barrientos, "Heterogeneous multi-robot system for mapping environmental variables of greenhouses," *Sensors*, vol. 16, no. 7, p. 1018, 2016.

[13] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4775–4782.

[14] G. Hitz, A. Gotovos, M.-É. Garneau, C. Pradalier, A. Krause, R. Y. Siegwart, *et al.*, "Fully autonomous focused exploration for robotic environmental monitoring," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2658–2664.

[15] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, ser. AAMAS '14. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014, p. 573–580.

[16] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Lió, and K.-W. Lee, "Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment," *Ad Hoc Networks*, vol. 7, no. 4, pp. 725–741, 2009.

[17] G. Notomista, S. F. Ruf, and M. Egerstedt, "Persistification of robotic tasks using control barrier functions," *Robotics and Automation Letters*, vol. 3, no. 2, pp. 758–763, 2018.

[18] P. Salaris, M. Cognetti, R. Spica, and P. Robuffo Giordano, "Online optimal perception-aware trajectory generation," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1307–1322, 2019.

[19] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[20] D. P. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[21] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation, and control*. Routledge, 2018.

[22] E. D. Sontag, "A lyapunov-like characterization of asymptotic controllability," *SIAM journal on control and optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[23] ——, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.

[24] R. A. Freeman and J. A. Primbs, "Control lyapunov functions: New ideas from an old source," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 4. IEEE, 1996, pp. 3926–3931.

[25] J. A. Primbs, V. Nevistić, and J. C. Doyle, "Nonlinear optimal control: A control lyapunov function and receding horizon perspective," *Asian Journal of Control*, vol. 1, no. 1, pp. 14–24, 1999.

[26] J. Crank, *The mathematics of diffusion*. Oxford university press, 1979.

[27] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.