

Decentralized Trajectory Optimization for Multi-Agent Ergodic Exploration

Dimitris Gkouletsos, Andrea Iannelli, Mathias Hudoba de Badyn, John Lygeros

Abstract—Autonomous exploration is an application of growing importance in robotics. A promising strategy is ergodic trajectory planning, whereby an agent spends in each area a fraction of time which is proportional to its probability information density function. In this paper, a decentralized ergodic multi-agent trajectory planning algorithm featuring limited communication constraints is proposed. The agents’ trajectories are designed by optimizing a weighted cost encompassing ergodicity, control energy and close-distance operation objectives. To solve the underlying optimal control problem, a second-order descent iterative method coupled with a projection operator in the form of an optimal feedback controller is used. Exhaustive numerical analyses show that the multi-agent solution allows a much more efficient exploration in terms of completion task time and control energy distribution by leveraging collaboration among agents.

Index Terms—Optimization and Optimal Control, Path Planning for Multiple Mobile Robots, Task and Motion Planning.

I. INTRODUCTION

IN recent years, autonomous exploration has received significant attention in view of a variety of application fields such as agriculture surveillance [1] and active map searching for disaster environments [2]. A promising framework in which to set the problem is ergodic trajectory planning [3], whereby the generated trajectory samples a region in the search space proportional to the expectation of how informative the region will be. Intuitively, an ideal ergodic trajectory should cover high-valued information regions proportionally to the time spent in that region. A projection-based iterative optimization algorithm was proposed in [4] to plan ergodic trajectories of a single-agent system. Experimental results [5] showcased the advantages of this approach, which outperformed alternative entropy minimization and information maximization strategies. In [6], an extension was presented for constrained environments and obstacle avoidance problems, and the Kullback-Leibler divergence was adopted as alternative to the ergodic metric.

Recent advances in computational resources and maturity of distributed algorithms have enabled multi-agent network collaborations in a variety of critical applications such as object

secure transportation [7] and search-and-prosecute missions [8]. Multi-agent configurations have been shown to outperform single-agent ones in terms of completion mission time [8], [9]. Decentralized solutions are of great appeal, since replacing a central unit by a network of agents planning in parallel the task execution gives more robustness against failures and provides more communication flexibility [10].

In this paper, we present a multi-agent decentralized trajectory planning problem in the framework of ergodic exploration. Related works are: [11], which reformulates the single-agent problem in [4] using a Nash equilibrium interpretation for the multi-agent setting; [12], focusing on area coverage with obstacles; and [13] which demonstrates a decentralized ergodic swarm control framework adaptable to external user commands and dynamic environmental information. The contribution of this work is twofold. First, a decentralized multi-agent extension of the approach in [4] is proposed (Section III). The algorithm consists of four steps all performed at agent-level: a second-order steepest descent optimizer, combined with a line-search scheme, determines a candidate trajectory that is optimal according to a generalized global cost function (discussed in III); a feasible trajectory that satisfies the dynamic constraints of the agent is obtained by projection [14]; estimates of the other agents trajectories are updated by averaging. In contrast to previous works, the cost determining the optimal trajectory encompasses: a (global) ergodic metric; a (global) control energy index; and a penalty on the inter-agent distance. This new cost definition, and the applicability of the approach to general nonlinear systems without requiring control affine dynamics, are important differences with respect to [11], where the only objective was to optimize ergodicity and thus could make use of a different algorithmic approach. Second, differently from previous works, we provide a systematic investigation of the benefits of multi-agent systems against single-agent systems in the ergodic exploration framework (Section V). Even though experimental results are not provided here, it is believed that the reported numerical analyses and the insights gained therein represent an invaluable starting point to plan a future experimental study, similar in concept to the one reported in [5] for the single-agent case. A range of performance metrics, including completion task time and control energy, are reported for different numbers of agents and network topologies randomly generated. An empirical study is also performed to describe the convergence property of the algorithm. Section IV details all the practical aspects of the algorithm for the considered scenarios and accompanies the release of the repository [15] that reproduces the results.

Manuscript received: February, 24, 2021; Revised May, 26, 2021; Accepted June, 22, 2021.

This work has been partially supported by the Swiss National Science Foundation under NCCR Automation.

The authors are with the Department of Information Technology and Electrical Engineering, Automatic Control Lab, ETH, Zürich 8092, Switzerland. (corresponding author: Andrea Iannelli) emails: dgkouletsos@student.ethz.ch, {iannelli, mbadyn, lygeros}@control.ee.ethz.ch

Digital Object Identifier (DOI): 10.1109/LRA.2021.3094242.

II. ERGODICITY

A cursory overview of the ergodic trajectory planning from [3] is presented. Consider a rectangular domain $\mathcal{X} := [0, L_1] \times \dots \times [0, L_n]$ and an associated density of information formulated by a probability density function $p(\chi)$, with $\chi \in \mathcal{X}$. For an horizon T , denote by: $x(t) : [0, T] \rightarrow \mathbb{R}^n$ the state; $u(t) : [0, T] \rightarrow \mathbb{R}^m$ the control; and $\dot{x}(t) = f(x(t), u(t))$ the system dynamics. The following distribution gives information on the time-averaged statistics of a trajectory in \mathcal{X}

$$\mathcal{C}(\chi, x) = \frac{1}{T} \int_0^T \delta(\chi - x(\tau)) d\tau, \quad (1)$$

where $\delta(\cdot)$ is a Dirac delta function. Fourier series representations of $p(\chi)$ and $\mathcal{C}(\chi, x)$ are obtained by making use of the basis functions

$$F_k(\chi) = \frac{1}{h_k} \prod_{i=1}^n \cos\left(\frac{k_i \pi}{L_i} \chi_i\right), \quad (2)$$

where: $k = (k_1, k_2, \dots, k_n) \in \mathbb{Z}^n$ denotes a multi-index that belongs to the set $\mathcal{K} = \{k \in \mathbb{Z}^n | 0 \leq k_j \leq K_j\}$ with K_j referring to the highest selected cosine harmonic; and h_k is a normalization factor that guarantees that the Fourier basis functions F_k have unit norm. The Fourier coefficients of the spatial and time-averaged distributions, respectively p_k and c_k , can then be obtained through a standard inner product over the exploration domain \mathcal{X} . Figure 1 shows the conditions for ergodicity of a trajectory x with respect to subsets N_1 and N_2 (level sets of $p(\chi)$ are also reported).

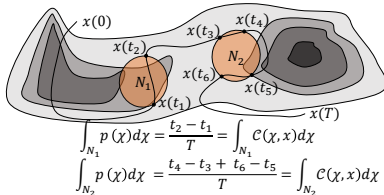


Fig. 1: Visualization of the conditions for ergodicity.

In the multi-agent case, superscripts are used to denote the agents index (e.g. $x^{(j)}$ is the state of the j -th agent) and $x = (x^{(1)}, \dots, x^{(N)})$ and $u = (u^{(1)}, \dots, u^{(N)})$ denote the stacked state and control vectors, where N is the number of agents. A global trajectory Fourier coefficient $C_k(x)$ that accounts for the effect of all agents trajectories is defined as

$$C_k(x) = \frac{1}{N} \sum_{j=1}^N c_k(x^{(j)}). \quad (3)$$

The *shared* ergodic metric, playing a key role in the problem formulation, captures the ergodicity of the multi-agent configuration as the weighted squared difference between the spatial and the time-averaged trajectory distributions via the respective Fourier coefficients

$$\mathcal{E} = \sum_{k \in \mathcal{K}} \Lambda_k (C_k(x) - p_k)^2, \quad (4)$$

where $\Lambda_k = (1 + \|k\|^2)^{-\frac{\lambda+1}{2}}$ and $\lambda \leq n$ is the number of exploratory variables in \mathcal{X} .

III. DECENTRALIZED ERGODIC EXPLORATION

This section presents the proposed decentralized algorithm for ergodic trajectory planning, which extends the single-agent problem formulation from [4] to a multi-agent setting described by undirected and connected network topologies.

Let us denote by $\xi^{(j)} := (\alpha^{(j)}, \mu^{(j)})$ a *planning trajectory* of the j -th agent; that is, a state-control pair (with $\alpha(t) : [0, T] \rightarrow \mathbb{R}^n$ and $\mu(t) : [0, T] \rightarrow \mathbb{R}^m$) which do not necessarily satisfy the system dynamics. Let us also denote by $\eta^{(j)} := (x^{(j)}, u^{(j)})$ a *feasible trajectory* of the j -th agent, that is a state-control pair belonging to the manifold \mathcal{T} of trajectories satisfying $\dot{x}^{(j)} = f(x^{(j)}, u^{(j)})$. See [16] for a formal characterization of \mathcal{T} . We also define $\xi = (\xi^{(1)}, \dots, \xi^{(N)})$ and $\eta = (\eta^{(1)}, \dots, \eta^{(N)})$ the augmented vectors consisting of the trajectories of all agents.

The problem is formulated as the minimization of an objective function featuring three contributions

$$J(\xi) = \underbrace{q \sum_{k \in \mathcal{K}} \Lambda_k (C_k(\alpha) - p_k)^2}_{\text{ergodicity}} + \underbrace{\int_0^T \sum_{j=1}^N \frac{1}{2} \|\mu^{(j)}(\tau)\|_{R(\tau)}^2 d\tau}_{\text{control energy}} + \underbrace{\int_0^T \sum_{j=1}^N \sum_{\ell=j+1}^N \frac{1}{r_{j\ell}(\tau) + \frac{1}{2} \|\alpha^{(j)}(\tau) - \alpha^{(\ell)}(\tau)\|_{W_{j\ell}(\tau)}^2 d\tau}_{\text{inter-agent distance}}, \quad (5)$$

with the following design parameters: $q > 0 \in \mathbb{R}$ penalizes ergodicity; $R(\cdot) \succeq 0 \in \mathbb{S}^{m \times m}$ is a (time-varying) penalty for the control energy; $r_{j\ell}(\cdot) > 0 \in \mathbb{R}$ is a (time-varying) penalty for the inter-agent distance; $W_{j\ell}(\cdot) \succeq 0 \in \mathbb{S}^{n \times n}$ is a linear transformation that allows distance between two agents positions to be computed. The inter-agent distance cost encourages each agent, via the choice of $r_{j\ell}$, to perform exploration while avoiding collisions with the others. The convention $\|x\|_Q^2 = x^\top Q x$ will be used throughout.

The goal is to determine planning trajectories $\xi^{(j)} \in \mathcal{T}$ that minimize (5). Following [14], the nonlinear constraint imposed by the trajectory manifold \mathcal{T} is removed by making use of a projection operator \mathcal{P} , which maps planning trajectories ξ to feasible trajectories η . That is, the following optimization problem is considered

$$\min_{\xi} J\left(\mathcal{P}\left(\xi^{(1)}\right), \dots, \mathcal{P}\left(\xi^{(N)}\right)\right), \quad (6)$$

and the optimal trajectory is taken as $\eta^* = \mathcal{P}(\xi^*)$, where ξ^* is a minimizer of (6). Problem (6) is non-convex, and thus a local minimizer ξ^* is sought. To this end, given information density Fourier coefficients p_k and initial candidate trajectories ξ_0 , (6) is solved via an iterative steepest descent algorithm, whereby each agent optimizes, in parallel and only using information from neighbouring agents, its own trajectory with the goal of minimizing the global cost (5). The iterative algorithm consists of four steps, which are the topic of the next subsections.

A. Step one: steepest descent

All agents maintain estimates of the trajectories of all other agents, thus denote by $\{\eta^{(\ell)}\}_j$ the feasible trajectory of the ℓ -th agent estimated by the j -th agent, and by $\{\eta\}_j =$

$(\{\eta^{(1)}\}_j, \dots, \{\eta^{(N)}\}_j)$ the stacked vector. Same convention is used for the planning trajectories ξ .

At iteration i , each agent determines its descent direction $\zeta_i^{(j)}$ to minimize locally the objective function J . To this end, the objective function J is approximated by a second-order Taylor expansion around $\xi_i^{(j)}$

$$\min_{\zeta_i^{(j)} \in T_{\xi_i^{(j)}} \mathcal{T}} D_{\xi_i^{(j)}} J(\{\xi_i\}_j) \circ \zeta_i^{(j)} + \frac{1}{2} (\zeta_i^{(j)}, \zeta_i^{(j)}), \quad (7)$$

where $T_{\xi_i^{(j)}} \mathcal{T}$ is the tangent trajectory manifold of \mathcal{T} , $D_{\xi_i^{(j)}} J(\{\xi_i\}_j) \circ \zeta_i^{(j)}$ is the first Frechet directional derivative and $(\zeta_i^{(j)}, \zeta_i^{(j)})$ is a quadratic approximation of the second Frechet directional derivative. The constraints force the direction $\zeta_i^{(j)}$ to lie on the tangent space of the trajectory manifold \mathcal{T} [14]. The direction $\zeta_i^{(j)} = (z_i^{(j)}, v_i^{(j)})$ is divided into state direction $z_i^{(j)}$ and control direction $v_i^{(j)}$. For clarity, we omit the iteration subscript i and assume that a feasible trajectory $\eta^{(j)}$ is available (this is natural considering that projection operator \mathcal{P} is applied at the end of each iteration).

1) *First Frechet directional derivative* : The first Frechet directional derivative $D_{\eta^{(j)}} J(\{\eta\}_j) \circ \zeta^{(j)}$ can be written as

$$\begin{aligned} & D_{x^{(j)}} J(\{\eta\}_j) \circ z^{(j)} + D_{u^{(j)}} J(\{\eta\}_j) \circ v^{(j)} \\ &= \int_0^T a_j(\tau)^\top z^{(j)}(\tau) d\tau + \int_0^T b_j(\tau)^\top v^{(j)}(\tau) d\tau, \end{aligned} \quad (8)$$

where

$$\begin{aligned} a_j(\tau) &= \sum_{k \in \mathcal{K}} \frac{2q\Lambda_k}{N \cdot T} \left(C_k(\{x\}_j) - p_k \right) \nabla_{x^{(j)}} F_k(x^{(j)}(\tau)) \\ &+ \sum_{i=1}^N \frac{-W_{ji}(\tau) \left(x^{(j)}(\tau) - \{x^{(i)}(\tau)\}_j \right)}{\left(r_{ji}(\tau) + \frac{1}{2} \left\| x^{(j)}(\tau) - \{x^{(i)}(\tau)\}_j \right\|_{W_{ji}(\tau)}^2 \right)^2}. \end{aligned} \quad (9)$$

and

$$b_j(\tau) = R(\tau)u^{(j)}(\tau). \quad (10)$$

2) *Second Frechet directional derivative*: A quadratic approximation of $(\zeta^{(j)}, \zeta^{(j)})$ can be obtained as

$$\int_0^T \left(\|z^{(j)}(\tau)\|_{Q_n(\tau)}^2 + \|v^{(j)}(\tau)\|_{R_n(\tau)}^2 \right) d\tau + \|z^{(j)}(T)\|_{P_{1n}}^2 \quad (11)$$

where $Q_n(\cdot) \in \mathbb{R}^{n \times n}$ and $P_{1n} \in \mathbb{R}^{n \times n}$ are positive semi-definite and $R_n(\cdot) \in \mathbb{R}^{m \times m}$ is positive definite.

3) *Optimization*: Using (8) and (11), the optimization problem to find the descent direction for the trajectory update can be formulated as

$$\begin{aligned} & \min_{(z^{(j)}, v^{(j)})} \int_0^T \left(a_j(\tau)^\top z^{(j)}(\tau) + b_j(\tau)^\top v^{(j)}(\tau) \right. \\ & \left. + \frac{1}{2} \|z^{(j)}(\tau)\|_{Q_n(\tau)}^2 + \frac{1}{2} \|v^{(j)}(\tau)\|_{R_n(\tau)}^2 \right) d\tau + \frac{1}{2} \|z^{(j)}(T)\|_{P_{1n}}^2 \\ \text{s.t. } & \dot{z}^{(j)} = \frac{\partial f}{\partial x^{(j)}} z^{(j)} + \frac{\partial f}{\partial u^{(j)}} v^{(j)}. \end{aligned} \quad (12)$$

where the linearized dynamics constraint enforces that $\zeta^{(j)} \in T_{\xi^{(j)}} \mathcal{T}$. The optimization problem in (12) provides an optimal descent direction [14] and can be solved via differential Riccati equations [17].

B. Step two: Armijo line search

To update the current planning trajectory $\xi_i^{(j)}$ with respect to the descent direction $\zeta_i^{(j)}$, we adopt the common Armijo rule [18]. The optimal step-size is defined as

$$\begin{aligned} & \max_{\gamma_i^{(j)} \in (0,1]} \gamma_i^{(j)} \\ \text{s.t. } & J(\dots, \xi_i^{(j)} + \gamma_i^{(j)} \zeta_i^{(j)}, \dots) - J(\{\xi_i\}_j) \\ & \leq \rho \cdot \gamma_i^{(j)} \cdot D_{\xi_i^{(j)}} J(\{\xi_i\}_j) \circ \zeta_i^{(j)}. \end{aligned} \quad (13)$$

where the user-defined parameter ρ sets the desired magnitude decrease required to achieve an adequate objective improvement. It is important to observe that the Armijo rule needs global information to guarantee the cost function decrease. Here, in line with the approximation proposed for step one, each agent implements a local version of the rule using the current iterates of the neighbours and the estimates of the trajectories of all other agents. This is a heuristic, whose sub-optimality we intend to analyse in future research. A starting point could be extending the local Armijo rule for strictly convex and separable cost functions with linear constraints developed in [19] to the non-convex setting analyzed in this paper.

C. Step three: projection on feasible trajectories manifold

The projection operator \mathcal{P} maps a planning trajectory into the closest one belonging to the trajectory manifold \mathcal{T} made of trajectories that satisfy $\dot{x}^{(j)} = f(x^{(j)}, u^{(j)})$, that is $\mathcal{P} : \xi^{(j)} = (\alpha^{(j)}, \mu^{(j)}) \mapsto \eta^{(j)} = (x^{(j)}, u^{(j)})$. It was shown in [14] that the projection operator \mathcal{P} can be interpreted as the following trajectory tracking controller

$$\eta^{(j)} = \mathcal{P}(\xi^{(j)}) : \begin{cases} u^{(j)} = \mu^{(j)} + K^{(j)}(\alpha^{(j)} - x^{(j)}) \\ \dot{x}^{(j)} = f(x^{(j)}, u^{(j)}) \end{cases} \quad (14)$$

The optimal controller gain $K^{(j)}$ can be computed as the solution of a finite horizon Linear Quadratic Regulator problem [17] applied to a linearization of the nonlinear dynamics around the trajectory $\xi^{(j)} = (\alpha^{(j)}, \mu^{(j)})$. A useful criterion for choosing the weighting matrices $R_{LQR} \in \mathbb{R}^{m \times m}$ and $Q_{LQR} \in \mathbb{R}^{n \times n}$ is to tune them so that the projected trajectory $\eta^{(j)}$ is close to $\xi^{(j)}$ and thus the linearization gives sufficiently accurate results.

D. Step four: agents trajectories estimation

Agent j has only access to the trajectories optimized by the agents in its neighbourhood $\mathcal{N}(j)$. At the end of round i ,

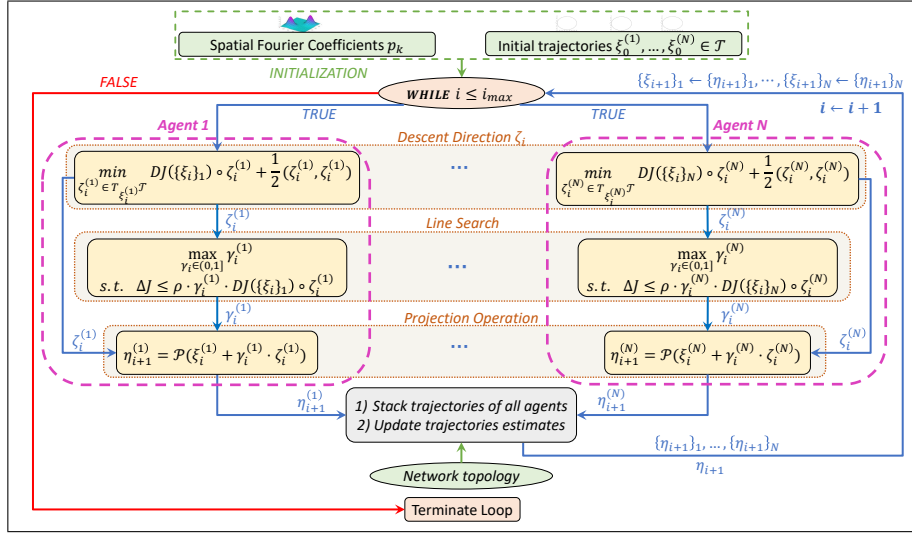


Fig. 2: Flowchart of the proposed multi-agent decentralized ergodic trajectory optimization algorithm.

the vector of feasible trajectories estimated by the j -th agent $\{\eta^{(\ell)}\}_j$ is obtained by the communication protocol below

$$\{\eta_i^{(\ell)}\}_j = \begin{cases} \eta_i^{(\ell)} & \text{if } \ell \in \{\mathcal{N}(j) \cup \{j\}\} \\ \frac{1}{|\mathcal{N}(j)| + 1} \left(\sum_{k \in \mathcal{N}(j) \cup \{j\}} \{\eta_{i-1}^{(\ell)}\}_k \right), & \text{otherwise.} \end{cases} \quad (15)$$

If an agent ℓ is not included in the neighbourhood of the j -th agent, then the j -th agent averages its estimate of agent ℓ 's trajectories with its neighbours' estimates of agent ℓ 's trajectory. Figure 2 illustrates a flowchart of the iterative optimization algorithm.

The termination criterion consists of stopping the algorithm after a maximum number of iterations i_{max} has been performed. Because the algorithm is run offline, the value of i_{max} can be chosen large enough so that the *ergodic reduction* metric $\mathcal{E}_r := 100 \frac{\mathcal{E}_0 - \mathcal{E}_f}{\mathcal{E}_0}$, where \mathcal{E}_f and \mathcal{E}_0 are the ergodicity at the i_{max} -th and initial trajectory, respectively, has a satisfactory value. Alternatively, each agent can broadcast to its neighbours a flag when its local problem has reached the termination criterion (which can include e.g. a local estimate of \mathcal{E}_r and the local directional derivative), which in subsequent iterations is rebroadcast to their (the neighbours') neighbours, and so on. Once an agent has collected termination flags from all other agents, it ceases running the algorithm. This happens G iterations after all agents have reached their respective local termination criteria, where G is the length of the longest path in the graph (or *girth*).

IV. TEST-CASE PRESENTATION

The complete model description, the algorithm's parameters and the performance metrics used to evaluate the proposed algorithm are provided in this section.

A. Agents dynamics

We consider the nonlinear dynamic model for the motion of the single agent used in [4] and a time horizon $T = 3.5$ sec. That is

$$\dot{x}^{(j)} = f(x^{(j)}, u^{(j)}) = \begin{bmatrix} \cos(\theta^{(j)}) & 0 \\ \sin(\theta^{(j)}) & 0 \\ 0 & 1 \end{bmatrix} u^{(j)}. \quad (16)$$

The state vector is $x^{(j)} = [X^{(j)}, Y^{(j)}, \theta^{(j)}]^\top$ where $X^{(j)}$ and $Y^{(j)}$ correspond to Cartesian coordinates, while $\theta^{(j)}$ is the heading angle of the velocity vector. The control input vector $u^{(j)} = [\nu^{(j)}, \omega^{(j)}]^\top$ consists of the forward velocity $\nu^{(j)}$ and the time derivative $\omega^{(j)}$ of the heading angle $\theta^{(j)}$. The initial feasible trajectories are circles with radius $\mathcal{R} = 0.05$ m and center randomized as discussed later.

B. Exploration field

The exploration field is assumed to be a two-dimensional space $[0, 1] \times [0, 1]$ corresponding to X and Y Cartesian coordinates and, thus, the number of exploratory variables is $\lambda = 2$. The information density, assigned to this field, is modelled based on a Gaussian mixture structure as follows

$$p(\chi) = \sum_{i=1} w_i \frac{1}{\sqrt{2\pi} |\Sigma_i|} e^{-\frac{1}{2} (M\chi - \mu_i)^\top \Sigma_i^{-1} (M\chi - \mu_i)}, \quad (17)$$

where: $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is a transformation that maps χ to the exploration variables; $\mu_i \in \mathbb{R}^2$ is the mean vector of the i -th mode; $\Sigma_i \in \mathbb{R}^{2 \times 2}$ is the positive definite covariance of the i -th mode; and $w_i \in [0, 1]$ is the weight of the i -th mode (chosen such that p integrates to 1 in the exploration domain). To investigate the ability of the algorithm to design different planning strategies as a function of the information densities, we investigate two different cases for (17). The first, named *volcano*, has a dominant mode at the center of the exploration map and other minor modes peripherally. The first mode has $w_1 = 0.6$, $\Sigma_1 = 0.014 \cdot \mathcal{I}_{2 \times 2}$ and $\mu_1 = [0.5 \ 0.5]^\top$,

whereas the rest of the modes are equally weighted with $\Sigma_i = 0.004 \cdot \mathcal{I}_{2 \times 2}$, $\mu_2 = [0.75 \ 0.5]^\top$, $\mu_3 = [0.25 \ 0.5]^\top$, $\mu_4 = [0.5 \ 0.75]^\top$ and $\mu_5 = [0.5 \ 0.25]^\top$. The second, named *archipelago*, has four modes with same covariance $\Sigma = 0.006 \cdot \mathcal{I}_{2 \times 2}$ and mean vectors $\mu_1 = [0.25 \ 0.25]^\top$, $\mu_2 = [0.75 \ 0.25]^\top$, $\mu_3 = [0.25 \ 0.75]^\top$, $\mu_4 = [0.75 \ 0.75]^\top$.

The approximation of spatial and trajectory distributions is addressed through the basis functions (2), where we set $K_1 = K_2 = 10$ for X and Y coordinates, respectively, and $K_3 = 0$ for θ coordinate as there is no exploration. It is noted that an increase in state dimension (16) has no effect on the computation of the coefficients, since one would assign zero to the indexes K_j associated with non-exploratory states j . Since the planning is done off-line, considering a higher order system would have little impact on the rest of the trajectory optimization problem.

C. Parameters and topology

Table I summarizes design parameters for the trajectory optimization algorithm. In the objective function, we prioritize ergodicity against control energy by adjusting accordingly the relative magnitudes of q and R . It can also be observed that matrix $W_{jl} = W$ extracts from the state vector elements related to Cartesian coordinates so that a Euclidean distance metric is obtained in the inter-agent cost term. The choice for Q_n and R_n is motivated by a desired smooth change on the state x and a more aggressive change on control u , respectively. The LQR weighting matrices have been tuned according to the previously discussed criterion.

TABLE I: Tuning parameters

Type	Parameters	Values
Objective Function J	q	100
	R	$0.03 \cdot \mathcal{I}_{2 \times 2}$
	W	$\text{diag}(1, 1, 0)$
Quasi-Newton	Q_n	$450 \cdot \mathcal{I}_{3 \times 3}$
	R_n	$14.5 \cdot \mathcal{I}_{2 \times 2}$
	P_{1n}	$50 \cdot \mathcal{I}_{3 \times 3}$
LQR	Q_{LQR}	$\mathcal{I}_{3 \times 3}$
	R_{LQR}	$\mathcal{I}_{2 \times 2}$

For the line search problem (13), we parameterize $\gamma_i^{(j)} = \beta^h$ with $\beta = 0.99$. With this choice, a fine search on the step-size is allowed and, thus, a larger reduction of the objective function is achieved, with convergence benefits [20]. Parameter ρ specifies the magnitude of reduction in the sufficient decrease condition and a typical value $\rho = 10^{-4}$ is used. Finally, the termination criterion is $i_{max} = 70$.

The analyzed network topologies for $3 \leq N \leq 10$ were randomly generated and are shown in Fig. 3. This topology defines the fixed neighbourhood $\mathcal{N}(\cdot)$ used in (15) and establishing the communication constraints among the agents.

D. Performance metrics

The performance of the algorithm is investigated based on the four metrics defined below (the first two are global, while the last two are at agent-level):

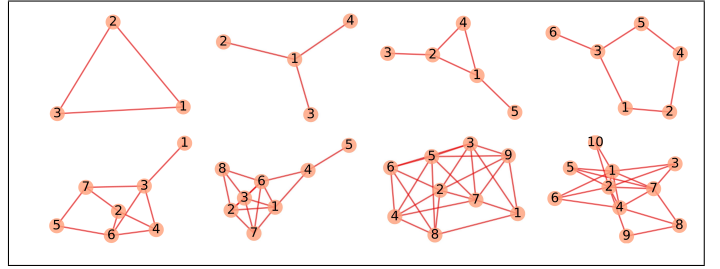


Fig. 3: Network topology for $3 \leq N \leq 10$.

- Optimal temporal ergodic metric [3]:

$$\mathcal{E}_{opt}(t) = \sum_{k \in \mathcal{K}} \Lambda_k (C_k(x^*, t) - p_k)^2, \quad (18)$$

where $C_k(x, t) = \frac{1}{N} \sum_{j=1}^N \frac{1}{t} \int_0^t F_k(x^{(j)}(\tau)) d\tau$. This is a time-dependent feature as it evaluates ergodicity with respect to any time instant in the horizon.

- Completion task time [9], [10]: it is generally used to indicate the time t_{CTT} at which exploration can terminate without loss of beneficial information. In this work, we define it as the first time instance $t \in [0, T]$ when $\frac{\mathcal{E}_{opt}(t=0) - \mathcal{E}_{opt}(t)}{\mathcal{E}_{opt}(t=0)} \geq \epsilon_{opt}$, where ϵ_{opt} is defined in Section V.
- Control energy per agent: $\sqrt{\int_0^{t_{CTT}} u^{(j)}(\tau)^2 d\tau}$.
- Traveled Distance: $\int_0^{t_{CTT}} |\nu^{(j)}(\tau)| d\tau$.

V. RESULTS

The implementation of the algorithm used to generate the results shown in this section are available at this repository [15].

A. Optimal trajectories

A comparison between the optimal trajectories in the Cartesian coordinates for single-agent (a) and 5-agent cases (b) with respect to the two information density distributions discussed earlier is presented in Figures 4 and 5. The figures also show the optimal ergodic metric $\mathcal{E}_{opt}(t)$ as a function of the time inside the horizon t . The inter-agent distance penalty term and optimal temporal ergodicity reduction tolerance are set to $r = 1$ and $\epsilon_{opt} = 99.5$, respectively. The initial trajectories used in the optimization are represented in dashed line, while the system's state initial condition x_0 at $t = 0$ is picked in low interest regions.

Figure 4a shows how the single-agent attempts to cover the whole region by spending more time in regions of higher interest, according to the ergodic principle. However, a high amount of time is required to cover efficiently the domain as shown by the completion task time value $t_{CTT} = 3.06$ sec. On the other hand, in the multi-agent case in Fig. 4b, the agents collaborate and split the field into exploration sub-domains. This is enabled by the use of a shared ergodic metric \mathcal{E} in the objective function (5) that frames exploration as a common group task among the agents. As a result, a more time-efficient exploratory mission compared to the single-agent case is accomplished (note that $t_{CTT} = 1.66$ sec value in Fig.

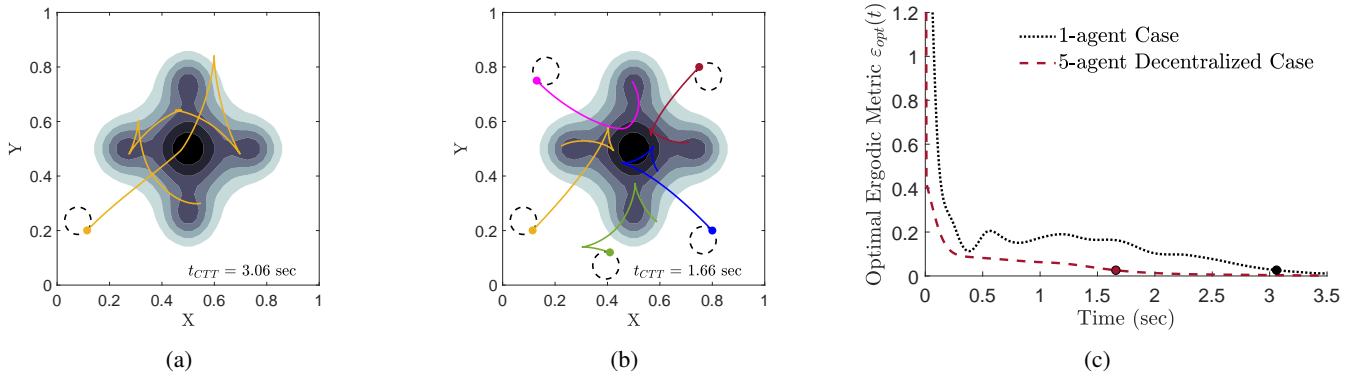


Fig. 4: Volcano case: State trajectory $(X(t), Y(t))$ for single agent (a) and five agents (b) from $t = 0$ until $t = t_{CTT}$; (c) $\mathcal{E}_{opt}(t)$ with respect to time. Dashed lines in (a) and (b) show the initial trajectory. Circular markers in (c) indicate t_{CTT} .

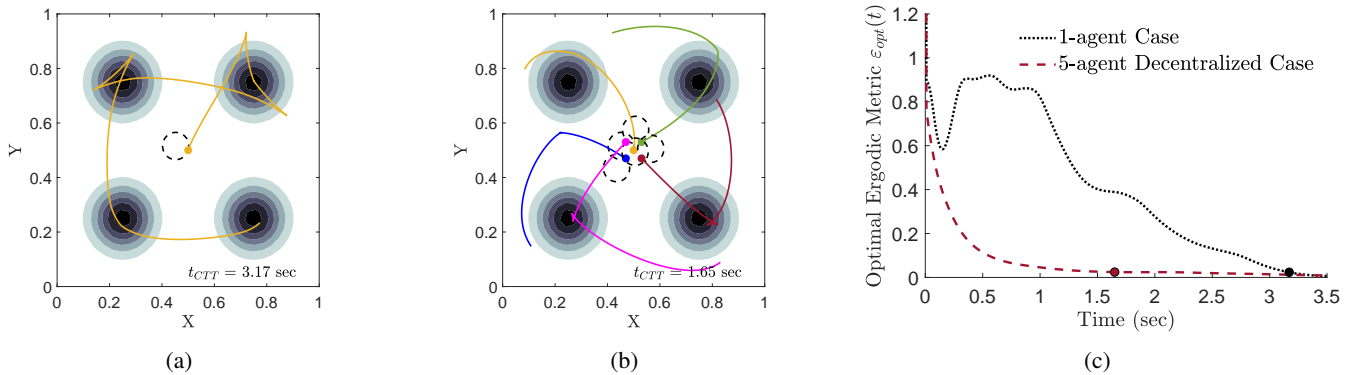


Fig. 5: Archipelago case: State trajectory $(X(t), Y(t))$ for single agent (a) and five agents (b) from $t = 0$ until $t = t_{CTT}$; (c) $\mathcal{E}_{opt}(t)$ with respect to time. Dashed lines in (a) and (b) show initial trajectory. Circular markers in (c) indicate t_{CTT} .

4b). Figure 4c, which displays the time-dependent optimal ergodic metric \mathcal{E}_{opt} , shows another distinctive feature of the multi-agent solution. Namely, \mathcal{E}_{opt} is monotonic with respect to time in the 5-agent case, unlike in the single-agent case. This can be explained by observing that the single-agent in Fig. 4a displays an initially efficient area coverage around high-valued information regions, but afterwards it spends a fraction of time in a low-valued information. This is necessary to move towards different high-valued regions, and determines the temporary increase of \mathcal{E}_{opt} . On the contrary, in the multi-agent case this inefficient part can be avoided by leveraging the possibility to optimize over multiple trajectories and thus distribute exploration to maximize the information reward. This feature, which had not been previously observed to the best of the authors knowledge, points out an additional benefit of the multi-agent configuration. Indeed, if \mathcal{E}_{opt} is monotonic, one is guaranteed to improve, in an ergodic sense, on learning the exploration field as time proceeds.

Figure 5 presents similar analyses for the Archipelago distribution. The single-agent visits all four modes spending more time in regions close to the peaks of the modes (Figure 5a), achieving a completion time ($t_{CTT} = 3.17$ sec) which is again very close to the horizon T of the mission. Collaborative planning is again observed for the 5-agent case in Fig. 5b, where it is noted that the five initial trajectories are chosen very

close to each other. Nonetheless, the optimized trajectories are distinct and allow the swarm to efficiently cover all modes ($t_{CTT} = 1.65$ sec). It is again instructive to observe the trend exhibited by the optimal ergodic metric \mathcal{E}_{opt} (Fig. 5c). In a qualitatively similar manner as before, the single-agent trajectory initially a decrease in ergodicity due to the coverage of the mode on the top-right. However, when it is directed to the second it spends a great amount of time in the intermediate low-valued information region, determining an over-shoot in the plot. This is clearly avoided in the multi-agent scenario configuration.

B. Quantitative aspects and initialization effects

We provide here a comprehensive analysis of the effect of increasing the number of agents and of choosing the initial conditions of the agents' state on three performance metrics. The number of considered agents is varied from 1 to 10, and, in each case, the optimization algorithm is run using 100 random initializations for the agents initial location x_0 . Precisely, we sample each initial condition from a uniform distribution such that $\theta(t = 0) \in [0, 2\pi]$, $X(t = 0) \in [0.05, 0.95]$ and $Y(t = 0) \in [0.05, 0.95]$. It is noted that this also randomizes the feasible trajectory used to initialize the optimization. As mentioned earlier, the trajectories are circles with given radius, and the selection of a point x_0 , together with the constraint that

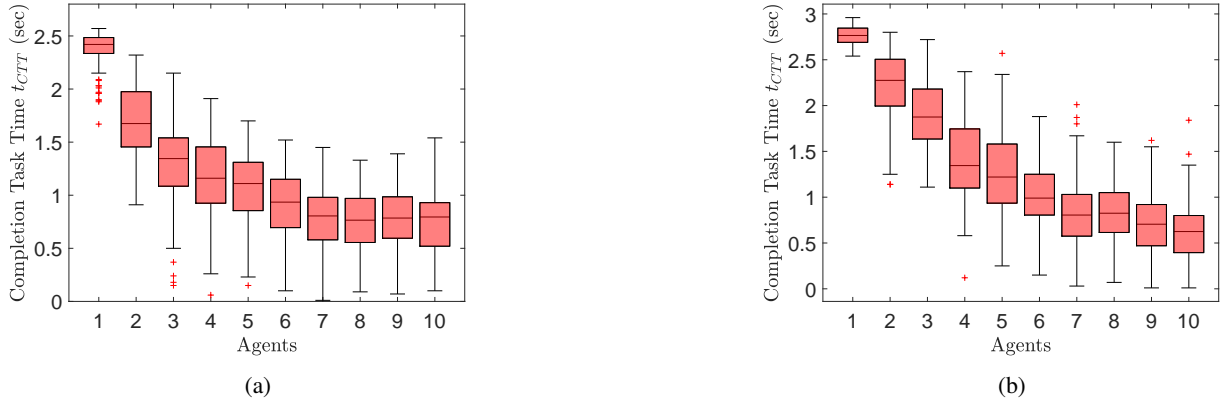


Fig. 6: Boxplot statistics of completion task time t_{CTT} versus number of agents for Volcano (a), Archipelago (b) distributions.

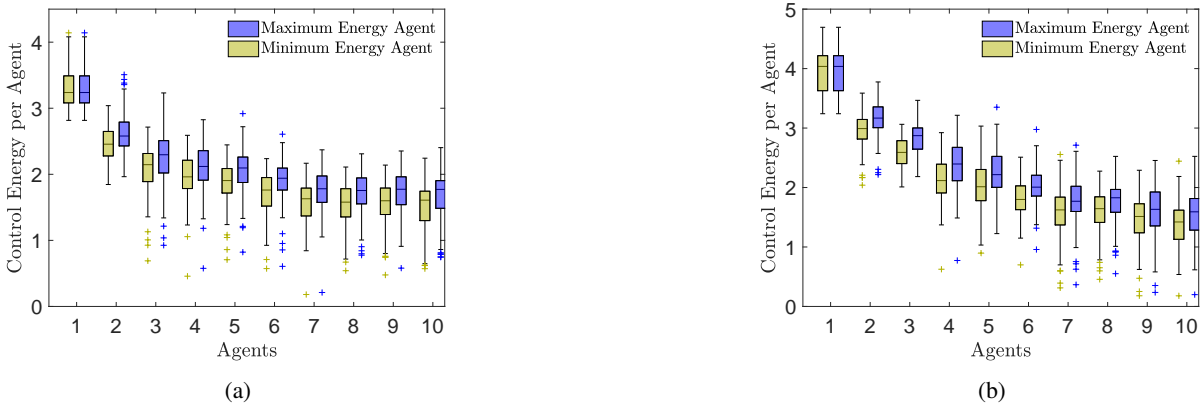


Fig. 7: Boxplot statistics of control energy per agent versus number of agents for Volcano (a), Archipelago (b) distributions.

the circular trajectory is feasible for the dynamics, uniquely determines its location in the field. These analyses thus shed also some light on the effect of the initial trajectory on the final result, which is an important aspect due to the non-convexity of the optimization problem.

Results are shown in terms of three performance metrics, namely the completion task time t_{CTT} , the control energy per agent, and the traveled distance \mathcal{D} . The inter-agent distance penalty term and optimal temporal ergodicity reduction tolerance are set to $r = 3$ and $\epsilon_{opt} = 98.5$, respectively.

Figures 6a and 6b show box plots of the completion task time statistics against number of agents for Volcano and Archipelago distributions, respectively. As expected, increasing the number of agents reduces the completion task time leading to more efficient exploration schemes. In the Volcano case, above 8 agents no further decrease is observed, suggesting that there is a distribution-dependent threshold for the largest number of agents giving an advantage in completion time. It is noted that 99.7% and 92.4% of the cases in Volcano and Archipelago distributions, respectively, have achieved an ergodicity reduction \mathcal{E}_r above 95%.

Figure 7 presents the control energy performance for the most and least energy consuming agents. This analysis highlights the advantageous distribution of energy for a multi-agent system. Indeed, by increasing the agents number, each

individual agent consumes lower energy and, thus, the energy pool is distributed efficiently among the agents.

Figure 8 finally shows the traveled distance box plot statistics. Recall that this metric is not explicitly targeted in the optimization, but it is nonetheless of practical interest to monitor it. As before, since this metric is a function of the agent, the minimum and maximum traveled distances are presented. Increasing the number of agents clearly leads to a decrease in the traveled distance. It is worth noting that it also leads to a decrease in the dispersion of this metric, both in terms of number of whiskers and box plots width.

C. Convergence study

Figure 9 reports the results of an investigation of the convergence properties of the decentralized algorithm. Specifically, the largest absolute value of the directional derivative across agents, i.e. $\max_{j \in \{1, \dots, N\}} |D_{\xi_i^{(j)}} J(\xi_i) \circ \zeta_i^{(j)}|$, normalized by the initial trajectory directional derivative at $i = 0$, is presented as a metric of optimality. The analyses are done on the graphs in Figure 3 for a randomly generated initial trajectory and initial condition x_0 .

A sublinear rate can be recognized in both examples. It is also observed that, probably due to the heuristic decentralization of the Armijo line search, there is no monotonic decrease

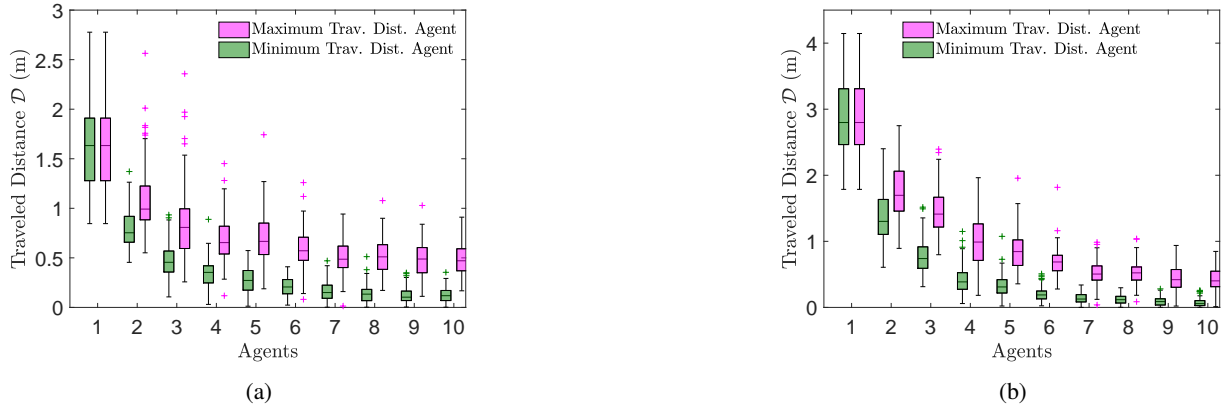


Fig. 8: Boxplot statistics of traveled distance \mathcal{D} versus number of agents for Volcano (a), Archipelago (b) distributions.

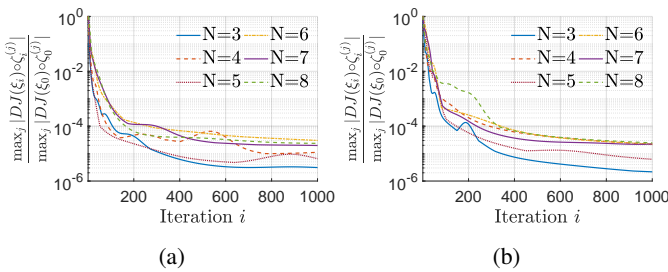


Fig. 9: Convergence of the directional derivative versus iteration rounds for Volcano (a) and Archipelago (b) distributions.

in the optimality metric. However, this does not compromise convergence. Analyses have also been carried out for two extreme network topologies, namely complete and line graphs, which showcased qualitatively similar trends (data not shown).

VI. CONCLUSION

The paper presents a new approach to design trajectories of multi-agent systems for ergodic exploration of stationary target distributions. To this aim, an objective function comprising three distinct terms is defined, and a decentralized optimization algorithm is proposed to minimize it. Two examples of distributions are considered in numerical experiments, and results are shown to demonstrate the validity of the approach and support the advantages of the proposed solution. The multi-agent algorithm enables more efficient exploration strategies compared to the single-agent case. Importantly, the shared ergodic metric allows multiple agents to explore cooperatively in order to search efficiently the domain with a low completion task time and a more efficient use of energy. Future work shall investigate an extension of the proposed method to scenarios where the target distribution can be updated online with measured data, as well as an experimental validation in a real-world environment.

REFERENCES

- [1] M. P. Christiansen, M. S. Laursen, R. N. Jørgensen, S. Skovsen, and R. Gislum, "Designing and testing a UAV mapping system for agricultural field surveying," *Sensors*, vol. 17, no. 12: 2703, 2017.
- [2] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.
- [3] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4-5, pp. 432–442, 2011.
- [4] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conference*, 2013.
- [5] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2015.
- [6] E. Ayvali, H. Salman, and H. Choset, "Ergodic coverage in constrained environments using stochastic trajectory optimization," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [7] H. Lee, H. Kim, and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 189–201, 2016.
- [8] J. G. Manathara, P. Sujit, and R. W. Beard, "Multiple UAV coalitions for a search and prosecute mission," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 1, pp. 125–158, 2011.
- [9] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413 – 14 423, 2020.
- [10] T. Nestmeyer, P. R. Giordano, H. H. Bühlhoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, p. 989–1011, 2017.
- [11] I. Abraham and T. D. Murphey, "Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2987 – 2994, 2018.
- [12] H. Salman, E. Ayvali, and H. Choset, "Multi-agent ergodic coverage with obstacle avoidance," in *International Conference on Automated Planning and Scheduling*, 2017.
- [13] A. Prabhakar, I. Abraham, A. Taylor, M. Schlafly, K. Popovic, G. Diniz, B. Teich, B. Simidchieva, S. Clark, and T. Murphey, "Ergodic specifications for flexible swarm control: From user commands to persistent adaptation," *Robotics: Science and Systems*, 2020.
- [14] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," in *15th IFAC World Congress*, 2002.
- [15] "Supplemental material to the paper "Decentralized Trajectory Optimization for Multi-Agent Ergodic Exploration"," DOI: 10.3929/ethz-b-000491536, 2021, ETH Research Collection.
- [16] A. Saccon, J. Hauser, and A. P. Aguiar, "Optimal Control on Lie Groups: The Projection Operator Approach," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2230–2245, 2013.
- [17] B. D. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Courier Corporation, 2007.
- [18] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [19] M. Zargham, A. Ribeiro, and A. Jadbabaie, "A distributed line search for network optimization," in *American Control Conference*, 2012.
- [20] C. T. Kelley, *Iterative Methods for Optimization*. SIAM, 1999.