

A Distributed Online Optimization Strategy for Cooperative Robotic Surveillance

Lorenzo Pichierri, Guido Carnevale, Lorenzo Sforini, Andrea Testa and Giuseppe Notarstefano

Abstract—In this paper, we propose a distributed algorithm to control a team of cooperating robots aiming to protect a target from a set of intruders. Specifically, we model the strategy of the defending team by means of an online optimization problem inspired by the emerging distributed aggregative framework. In particular, each defending robot determines its own position depending on (i) the relative position between an associated intruder and the target, (ii) its contribution to the barycenter of the team, and (iii) collisions to avoid with its teammates. We highlight that each agent is only aware of local, noisy measurements about the location of the associated intruder and the target. Thus, in each robot, our algorithm needs to (i) locally reconstruct global unavailable quantities and (ii) predict its current objective functions starting from the local measurements. The effectiveness of the proposed methodology is corroborated by simulations and experiments on a team of cooperating quadrotors.

I. INTRODUCTION

The employment of autonomous mobile robots for security purposes is becoming more and more important, see e.g. [1]–[4] and references therein. In this work, we investigate the framework in which a team of robots wants to protect a target (e.g., an asset or a region) from potential intruders.

Related Work: Several works address surveillance tasks in multi-robot settings by means of optimization procedures. Authors in [5]–[8] model the surveillance task as a coverage control problem. The work [9] proposes an algorithm to protect an asset by optimizing the positioning of a team of autonomous vehicles. In [10], authors propose a distributed, multi-objective algorithm to detect intruders and protect sensitive areas. In [11], [12], a team of autonomous vehicles is controlled to maximize the amount of time taken by an intruder to reach a protected asset. The work [13] addresses the problem of target defense by a team of water vehicles represented by learning rhythmic motor primitives. In [14], a dynamic, decentralized assignment algorithm based on one-to-one coordination is proposed to solve a multi-target allocation problem under communication constraints. In [15], a behavior-based fuzzy logic control system is presented to protect a target from potential threats. In [16], a surveillance task is addressed by leveraging Markov chains. Recently, the emerging distributed aggregative optimization

This result is part of the project “Distributed Optimization for Cooperative Machine Learning in Complex Networks” (No PGR10067) that has received funding from the Ministero degli Affari Esteri e della Cooperazione Internazionale.

Authors are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy. {lorenzo.pichierri, guido.carnevale, lorenzo.sforini, a.testa, giuseppe.notarstefano}@unibo.it.

framework has gained attention to model scenarios arising in cooperative robotics. In this field, a network of robots aims to minimize an objective function given by the sum of local functions which depend not only on a local decision variable (e.g., the position of the robot) but also on an aggregative quantity of the network (e.g., the barycenter of the team). This set-up has been introduced in the pioneering work [17]. Constrained, online versions of the problem have been investigated in [18], [19]. The authors of [20] consider communication with finite bits. In [21], a continuous-time distributed feedback optimization law is proposed to steer a set of single integrators to a steady-state configuration that is optimal with respect to an aggregative optimization problem. In [22], a distributed algorithm based on the Franke-Wolfe update has been proposed to reduce the computational effort.

Contributions: Inspired by the seminal works in [17], [18], in this paper we consider an online, constrained optimization problem over peer-to-peer networks of robots. Differently from previous approaches, we propose a novel optimization paradigm in which local cost functions depend not only on their own optimization variables but also on optimization variables associated to other robots. Moreover, we consider a challenging scenario tailored for multi-robot settings in which robots have access to local functions only by means of noisy measurements. This requires the introduction of an ad-hoc estimation procedure. Finally, aiming at reducing the dynamic regret, we propose a novel resolution strategy requiring a prediction step on local, unavailable quantities. To the best of the authors’ knowledge, this is the first work addressing such distributed setting. This novel optimization paradigm is applied to a cooperative, dynamic surveillance setting in which robots have to defend a target from a set of intruders while avoiding collisions. Finally, we apply the proposed resolution strategy to a team of real aerial robots mimicking a basket match.

Organization: The paper unfolds as follows. In Section II we introduce the dynamic multi-robot surveillance optimization problem. In Section III we detail how to model the optimization problem to fulfill the surveillance task. In Section IV we introduce the distributed resolution strategies. Simulations and experiments are provided in Section V.

Notation: I_n is the $n \times n$ identity. \otimes is the Kronecker product. $\text{blkdiag}(A, B)$ is the block diagonal concatenation of matrices A and B . Given a symmetric, positive-definite matrix $Q \in \mathbb{R}^{n \times n}$, and $x \in \mathbb{R}^n$, we define the Q -norm of x as $\|x\|_Q = \sqrt{x^\top Q x}$. Given a function $g: \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \rightarrow \mathbb{R}$, $\nabla_1 g(\cdot, \cdot) \in \mathbb{R}^{n_1}$ and $\nabla_2 g(\cdot, \cdot) \in \mathbb{R}^{n_2}$ are the gradient of g with respect to the first and the second argument, respectively.

II. PROBLEM FORMULATION

In this section, we detail the distributed online surveillance problem for a multi-robot cooperating team. Before introducing the considered optimization framework, we introduce the communication model connecting the cooperating robots.

A. Communication Model

In this paper, we consider a set $V = \{1, \dots, N\}$ of robots that have to accomplish a cooperative surveillance task. In this cooperative setting, robots have to leverage communications among each other in order to properly solve the problem. We assume that robots can exchange information according to a communication network modeled as a time-varying undirected graph $\mathcal{G}^t = (V, \mathcal{E}^t)$ in which $\mathcal{E}^t \subseteq V \times V$ is the edge set. A graph \mathcal{G}^t models the communication in the sense that there is an edge $(i, j), (j, i) \in \mathcal{E}^t$ if and only if robots i is able to exchange information with robots j at time t . We consider $(i, i) \in \mathcal{E}^t$ for each $t \geq 0$ and for each $i \in V$. We assume that robots are able to communicate if their relative distance is below a certain threshold. For each node i , the set of *neighbors* j such that there exists an edge $(i, j) \in \mathcal{E}^t$ of i at time t is denoted by $\mathcal{N}_i^t = \{j \in V \mid (i, j) \in \mathcal{E}^t\}$. Given $B > 0$, a graph is said to be *B-connected* if the graph $\{V, \bigcup_{\tau=t}^{t+B} \mathcal{E}^\tau\}$ is connected. We also associate a so-called weighted adjacency matrix $\mathcal{A}^t \in \mathbb{R}^{N \times N}$ matching the graph, i.e., a matrix with (i, j) -entry $a_{ij}^t > 0$ if $(i, j) \in \mathcal{E}$, otherwise $a_{ij}^t = 0$. The adjacency matrix is said to be doubly stochastic if $\sum_{i=1}^N a_{ij}^t = 1$ for all $j \in \{1, \dots, N\}$ and $\sum_{j=1}^N a_{ij}^t = 1$ for all $i \in \{1, \dots, N\}$.

In the next, we assume that, for all $t \geq 0$, the communication graph \mathcal{G}^t is *B-connected* and \mathcal{A}^t is doubly stochastic. Moreover, we also assume that there exists $a \in (0, 1)$ such that, for all $t \geq 0$ and $i, j \in \{1, \dots, N\}$, it holds $a_{ij}^t > a$ whenever $a_{ij}^t > 0$.

B. Distributed Online Optimization for Robotic Surveillance

The cooperative task consists of the surveillance of a certain, possible moving target. More in detail, we consider a dynamic scenario in which a team $I = \{1, \dots, N\}$ of intruding, adversarial robots moves towards the target (see, e.g., target surrounding problem in [17], [18]). Robots are modeled as generic discrete-time nonlinear systems. The generic robot $i \in V$ is aware of only one of the attackers $i \in I$. Moreover, it does not have any other information about other robots in V and I . To fulfill this defensive task, at the generic time $t \geq 0$ each robot in $i \in V$, cooperating with its teammates, chooses its configuration $x_i^t \in \mathbb{R}^3$ in the space according to the following policies:

- (i) stay between the attacker $i \in I$ and the target;
- (ii) stay near to other teammates and steer the barycenter of the team towards the target;
- (iii) avoid collisions with other robots in V ;
- (iv) move in a bounded area of the field.

An illustrative example of this set-up is provided in Figure 1. Here, each member of the defending team V , represented by blue spheres, wants to move between the

designated intruder and the target, represented as a flag. Moreover, the team of robots V is steering its barycenter towards the target.

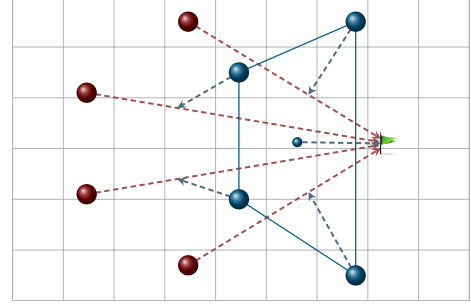


Fig. 1. Example of desired behavior. Each robot $i \in V$ wants to move in the line connecting the attacker $i \in I$ with the target (green flag). The barycenter (small blue sphere) is steered toward the flag.

We want to find the optimal configuration of the defending team with respect to the aforementioned tasks. To this end, we model this set-up by means of a distributed online optimization problem taking inspiration from the aggregative optimization framework [17]–[22]. In this optimization set-up, robots in V , at the generic time $t \geq 0$, aim at solving the following problem

$$\min_{x \in X^t} \sum_{i=1}^N f_i^t(x_i, \sigma(x), x_{\mathcal{N}_i^t}). \quad (1)$$

Here, $x = \text{col}(x_1, \dots, x_N)$, $x_{\mathcal{N}_i^t} = \{x_j\}_{j \in \mathcal{N}_i^t \setminus \{i\}}$, each $f_i^t(x_i, \sigma(x), x_{\mathcal{N}_i^t}) : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{3(|\mathcal{N}_i^t|-1)} \mapsto \mathbb{R}$ depends on the optimization variables of a subset of robots in V and on an *aggregative variable* $\sigma(x) \triangleq \sum_{i=1}^N \frac{\phi_i(x_i)}{N}$, with $\phi_i(x_i) : \mathbb{R}^3 \mapsto \mathbb{R}^3$ for each $i \in \{1, \dots, N\}$. We denote by $X^t = \prod_{i=1}^N X_i^t$ a constraint set on the position of robots at time t , where $X_i^t \subseteq \mathbb{R}^3$. The overall objective function is

$$f^t(x, \sigma(x)) = \sum_{i=1}^N f_i^t(x_i, \sigma(x), x_{\mathcal{N}_i^t}).$$

In the next section, we provide guidelines on how to shape the functions f_i^t , ϕ_i , and the sets X_i^t in order to affect the defensive behavior.

Throughout this paper, we consider a cooperative, distributed scenario in which robots have limited knowledge of the problem (1). Thus, in order to solve the optimization problem, they have to iteratively exchange suitable information via a communication network. In classical distributed online optimization frameworks [23], each robot $i \in V$ can only privately access f_i^t , ϕ_i , and X_i^t only once x_i^t has been computed. We instead consider a challenging scenario in which the robots in V have to estimate these quantities according to noisy measurements on the attackers and on the target. Moreover, in our setup, the quantities f_i^t and X_i^t are predicted before evaluating the optimal location estimate x_i^t .

III. MODELING

In this section, we detail a strategy to model the defensive behavior of the robotic team V within the considered

distributed online optimization setting. In the first part, we show how to model the cost function of the generic robot to surveil a certain adversary and how to model team behaviors for target protection and collision avoidance. In the second part of the section, we show how to properly constrain the movements of each robot. Finally, we provide details on how to predict the movements of the intruders. We do not consider any knowledge of the behavior of the offending team.

A. Modeling of Cost Function

In this section, we model the objective function of the generic robot i as in (1) as the sum of three terms, i.e., $f_i^t(x_i, \sigma(x), x_{N_i^t}) = f_{iS}^t(x_i) + f_{iC}^t(x_i, \sigma(x)) + f_{iB}^t(x_i, x_{N_i^t})$. More in detail, the first term $f_{iS}^t(x_i)$ models the behavior of the single robot and solely depends on the position x_i of the robot. The second term f_{iC}^t instead models a collective behavior and depends also on the position of other robots in the team V . The third term $f_{iB}^t(x_i, x_{N_i^t})$ instead handles collisions with other robots in V . We recall that the i^{th} robot does not have any knowledge of other robot positions. Moreover, robot i is only able to measure the position p_i^t of an intruder $i \in I$ at time t , thus not having knowledge of its future movements. The solution strategy detailed in Section IV will handle this lack of knowledge.

Each defending robot wants to track the associated offending intruder. Such defending strategy can be captured by defining for each robot i a local cost f_{iS}^t in the form $f_{iS}^t(x_i) = \|x_i - p_i^t\|_{Q_{i1}}^2$, where $Q_{i1} \in \mathbb{R}^{3 \times 3}$ is a symmetric, positive definite matrix. In this way, the robot is steered towards the adversary player.

In practical scenarios, it is desirable that the barycenter of the team V is near the target while robots surveil intruders. Moreover, robots may not want to scatter in the environment. Rather, they may want to be close to each other. Let $b^t \in \mathbb{R}^3$ be the position of the target at time t . Such defending strategy can be obtained by defining a local function f_{iC}^t of the form $f_{iC}^t(x_i, \sigma(x)) = \|\sigma(x) - b^t\|_{Q_{i2}}^2 + \|\sigma(x) - x_i\|_{Q_{i3}}^2$, where $Q_{i2}, Q_{i3} \in \mathbb{R}^{3 \times 3}$ are symmetric, positive definite matrices, while $\sigma(x)$ is designed to represent the center of mass of robots in V , i.e., $\sigma(x) = \frac{\sum_{i=1}^N x_i}{N}$.

In order to accomplish the collision avoidance task, the third term in the objective function is in the form $f_{iB}^t(x_i, x_{N_i^t}) = \sum_{j \in N_i^t \setminus \{i\}} \mathcal{B}(x_i, x_j)$ where (\cdot, \cdot) is some barrier function. A suitable example is

$$f_{iB}^t(x_i, x_{N_i^t}) = \sum_{j \in N_i^t \setminus \{i\}} -\log(\|x_i - x_j\|). \quad (2)$$

In practical settings, the generic robot i is not required to know the position x_j of other robots. Indeed, the distance $\|x_i - x_j\|$ and the relative distance vector $x_i - x_j$ can be retrieved, e.g. using Lidar sensors. Thereby, the function and its gradients can be evaluated using suitable measurements.

B. Modeling of Constraint Set

Let us introduce the constraints. We define $X_{\text{field}} \subseteq \mathbb{R}^3$ as the set of positions contained within the operating region. In

order to protect the target, we impose that each robot always stays between the intruder and the target, i.e., it must satisfy

$$\begin{cases} [p_i^t]_c + \epsilon_c^t \leq [x_i^t]_c \leq [b^t]_c & \text{if } [p_i^t]_c \leq [b^t]_c \\ [b^t]_c \leq [x_i^t]_c \leq [p_i^t]_c - \epsilon_c^t & \text{if } [p_i^t]_c > [b^t]_c \end{cases} \quad c = 1, 2, 3, \quad (3)$$

where $[\cdot]_c$ denotes the c -th component of the vectors and $\epsilon_c^t = \max(\epsilon_{c,\min}, \kappa_c |[p_i^t]_c - [b^t]_c|^2)$ is an additional tolerance with $\kappa_c > 0$ and a saturation term $\epsilon_{c,\min} > 0$. The overall feasible set X_i^t of each robot i at time t is then

$$X_i^t = X_{\text{field}} \cap \{x_i \in \mathbb{R}^3 \mid (3) \text{ is satisfied}\}. \quad (4)$$

We explain as follows the rationale behind the constraint sets X_i^t . Each defending robot $i \in V$ wants to stay between the corresponding intruder $i \in I$ and the target to be protected. The additional tolerances ϵ_c^t provide an operating margin that can be used by the defending robots to manage the fact that offending robots (and, possibly, the target) move over time. In particular, these tolerances are as small as the distance between the intruders and the target is small. Indeed, it is rather intuitive the fact that the defending robots need to stay closer to the intruders when they are close to the target. An illustrative 2D example of this constraint set is in Figure 2.

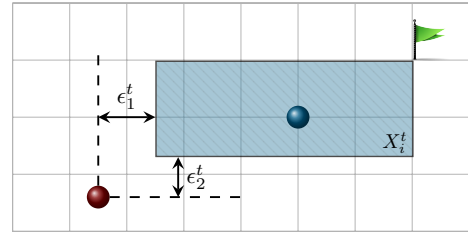


Fig. 2. Example of the constraint set for one of the robots $i \in V$. Robot i (blue sphere), can move in the constraint set as in (3) between the adversary (red sphere) and the target (green flag).

IV. SOLUTION STRATEGY

A. Prediction of Intruder and Target Behavior

As we detail in the next part of this section, in the proposed distributed strategy we rely on a prediction of target and adversarial moves. At the same time, since the precise intruder (or target) position could not be available at each time instant, e.g. the intruder is out of sight, it is necessary to determine the unknown position just by considering the preceding history. To overcome both these issues, and assuming sufficiently slow dynamics, we adopted a Kalman filter-based solution, [24].

For the sake of simplicity, we model each adversarial robot as a discrete-time double integrator system. Since we have no access to the input applied to each system, we approximate each member of the adversarial team as an autonomous system driven by a normally distributed signal, i.e.,

$$\begin{bmatrix} p_i^{t+1} \\ v_i^{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} I_3 & \Delta t I_3 \\ 0 & I_3 \end{bmatrix}}_{=:F} \begin{bmatrix} p_i^t \\ v_i^t \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \Delta t I_3 \end{bmatrix}}_{=:G} a_i^t \quad (5)$$

where $v_i^t \in \mathbb{R}^3$ is the velocity of the intruder i at time instant t , $\Delta t \in \mathbb{R}$ the discretization step and $a_i^t \in \mathbb{R}^3$ represents the effects of the unknown input applied to the i^{th} system. Specifically, we assume a_i^t drawn from a normal distribution with zero mean and covariance matrix $\sigma_i^2 I$. Let H be defined as $H := [I_3 \ 0]$. We assume we can access some measurements of the adversaries' states, namely, at each time t , we observe

$$z_{i,p}^t = H[(p_i^t)^\top, (v_i^t)^\top]^\top + w_{i,p}^t \quad (6)$$

where $z_{i,p}^t \in \mathbb{R}^3$ is the output of the i^{th} system and $w_{i,p}^t$ is a normally distributed disturbance with zero mean and covariance matrix $R_{i,p}$. Firstly, let us introduce some notation. We denote as $\xi_{i,p}^t$ the state of each system i at time t , i.e., $\xi_{i,p}^t := \text{col}(p_i^t, v_i^t)$. The estimation, computed at time t , of $\xi_{i,p}^{t+1}$ is denoted as $\hat{\xi}_{i,p}^{t+1}$. Moreover, to each estimate $\hat{\xi}_{i,p}^{t+1}$ we can associate a covariance matrix $P_{i,p}^{t+1} \in \mathbb{R}^{6 \times 6}$, which can be seen as a measure of the estimated accuracy of the current state estimate. We now recall the Kalman filter equations applied to our setting. Specifically, the approach is an iterative process consisting of two phases: prediction and correction. For all t , in the prediction phase a new state estimation (and covariance matrix), based on the system model is computed, namely,

$$\hat{\xi}_{i,p}^{t+1} = F \hat{\xi}_{i,p}^t \quad P_{i,p}^{t+1} = F P_{i,p}^t F^\top + S_{i,p} \quad (7)$$

where $S_{i,p} := G G^\top \sigma_i^2$. As for the correction phase, as soon as a measurement $z_{i,p}^t$ of $\xi_{i,p}^t$ is available (via (6)), we correct the estimation obtained by (7) via

$$\hat{\xi}_{i,p}^{t+1} = \hat{\xi}_{i,p}^{t+1} + K_i (z_{i,p}^t - H \hat{\xi}_{i,p}^{t+1}) \quad (8a)$$

$$P_{i,p}^{t+1} = (I_6 - K_i H) P_{i,p}^{t+1} (I_6 - K_i H)^\top + K_i R_{i,p} K_i^\top \quad (8b)$$

where K_i is the so-called Kalman gain computed as

$$K_{i,p} = F P_{i,p}^t H^\top (H P_{i,p}^t H^\top + R_{i,p})^{-1}. \quad (8c)$$

The updates (7) and (8) are performed iteratively with initial conditions $\xi_{i,p}^0 = \xi_{i,p}^{\text{init}}$, and $P_{i,p}^0 = P_{i,p}^{\text{init}}$. In general, $P_{i,p}^{\text{init}}$ can be seen as a measure of the accuracy of the initial state estimate $\xi_{i,p}^{\text{init}}$. The prediction-correction update can be written in its compact form

$$\hat{\xi}_{i,p}^{t+1} = (F - K_{i,p} H) \hat{\xi}_{i,p}^t + K_{i,p} z_{i,p}^t$$

$$P_{i,p}^{t+1} = (F - K_{i,p} H) P_{i,p}^t (F - K_{i,p} H)^\top + K_{i,p} R_{i,p} K_{i,p}^\top + S_{i,p}$$

At time t the intruder position can be estimated as $\hat{p}_i^t = H \hat{\xi}_{i,p}^t$. A similar approach is adopted to estimate the target position. That is, target dynamics is approximated as (cf. (5)), $\xi_b^{t+1} = F \xi_b^t + G a_b^t$, where $\xi_b^t := \text{col}(b^t, v_b^t)$ with b^t is the position of the target and v_b^t its velocity and a_b^t is sampled from a normal random variable with covariance matrix $\sigma_b^2 I$. Robot i accesses the noisy measurement $z_{i,b}^t = H \xi_b^t + w_{i,b}^t$ of ξ_b^t . The disturbance $w_{i,b}^t$ is normally distributed with zero mean and covariance matrix $R_{b,i}$. Each robot then computes an estimate $\hat{\xi}_{i,b}^t$ of ξ_b^t , with its covariance $P_{i,b}^t$, as in (7)–(8). Hence, the target position estimate \hat{b}_i^t computed by agent i can be obtained similarly as $\hat{b}_i^t = H \hat{\xi}_{i,b}^t$. The overall

procedure is implemented by each robot via Algorithm 1 where the following notation is adopted. Let us define $F := I_2 \otimes F$, $H := I_2 \otimes H$, $S_i := \text{blkdiag}(S_{i,p}, S_{i,b})$, $R_i := \text{blkdiag}(R_{i,p}, R_{i,b})$, $K_i := \text{blkdiag}(K_{i,p}, K_{i,b})$. Moreover, we denote as $P_i^t := \text{blkdiag}(P_{i,p}^t, P_{i,b}^t)$, $\hat{\xi}_i^t := \text{col}(\hat{\xi}_{i,p}^t, \hat{\xi}_{i,b}^t)$, $z_i^t := \text{col}(z_{i,p}^t, z_{i,b}^t)$ and $w_i^t := \text{col}(w_{i,p}^t, w_{i,b}^t)$.

B. Distributed Prediction and Optimization Scheme

The defending team runs Algorithm 1 (see the table below) to choose the positions x_i^t for all $t \geq 0$. The aim of the algorithm is to minimize the dynamic regret R_T , i.e., the performance metric $R_T := \sum_{t=1}^T (f^t(x^t, \sigma(x^t)) - f^t(x_\star^t, \sigma(x_\star^t)))$, where $T > 1$ denotes a given time horizon and $x_\star^t \in \mathbb{R}^3$ denotes a minimizer of $f^t(\cdot, \sigma(\cdot))$ over the set X^t . We underline that the objective functions introduced in Section III-A are strongly convex, and the constraint set described in Section III-B is non-empty, closed, and convex. Thus, the minimizer x_\star^t is unique for all t .

The algorithm structure described in Algorithm 1 relies on two main features. The first one regards a (local) tracking mechanism that allows each robot in the network to reconstruct global quantities that cannot be directly accessed. The second feature involves a (local) prediction mechanism giving rise to the current estimates of both the (local) objective function and the feasible set. Indeed, as usual in the context of distributed online optimization (see, e.g., the recent survey [23]), we assume that both f_i^{t+1} and X_i^{t+1} are revealed to robot i only once the update x_i^{t+1} has been computed. As a consequence, even with strongly convex objective functions, the existing bounds on the dynamic regret necessarily suffer the presence of terms $O(\sum_{t=1}^{T-1} \|x_\star^{t+1} - x_\star^t\|)$, [23]. We overcome this issue by implementing the estimation and prediction mechanism in Section IV-A, so that each robot i of the network is able to compute the estimates \hat{f}_i^{t+1} and \hat{X}_i^{t+1} of the objective function f_i^{t+1} and the feasible set X_i^{t+1} , respectively. We highlight that such a mechanism can be used thanks to the specific structure of the problem modeled in Section III. Indeed, both f_i^{t+1} and X_i^{t+1} could be fully computed assuming p_i^{t+1} and b^{t+1} known at time t and approximating f_{iB}^{t+1} with f_{iB}^t .

We now depict the main idea behind the steps (10a)–(10d) of Algorithm 1. In order to solve problem (1), one may implement a projected gradient method in a distributed way (see, e.g., [25]). Indeed, when applied to problem (1), the i^{th} block should read

$$x_i^{t+1} = P_{X_i^{t+1}} [x_i^t - \alpha [\nabla f^{t+1}(x^t, \sigma(x^t))]_i], \quad (9)$$

where $P_{X_i^{t+1}}[\cdot] \in \mathbb{R}^3$ denotes the projection operator over the set $X_i^{t+1} \subseteq \mathbb{R}^3$ and, with slight abuse of notation, $[v]_i \in \mathbb{R}^3$ denotes the i^{th} block in \mathbb{R}^3 of a given vector $v \in \mathbb{R}^{3N}$. However, we remark that the locations p_i^{t+1} and b^{t+1} are measured only once x_i^{t+1} has been computed and, hence, we can only use the predictions \hat{p}_i^{t+1} and \hat{b}_i^{t+1} to obtain \hat{f}^{t+1} and \hat{X}_i^{t+1} . Hence, we rewrite (9) as

$$x_i^{t+1} = P_{\hat{X}_i^{t+1}} [x_i^t - \alpha [\nabla \hat{f}^{t+1}(x^t, \sigma(x^t))]_i].$$

Now, by applying the chain rule we observe that

$$\begin{aligned} [\nabla \hat{f}^{t+1}(x^t, \sigma(x^t))]_i &= \nabla \hat{f}_{iS}^{t+1}(x_i^t) + \nabla_1 \hat{f}_{iC}^{t+1}(x_i^t, \sigma(x^t)) \\ &\quad + \nabla \phi_i(x_i^t) \frac{1}{N} \sum_{j=1}^N \nabla_2 \hat{f}_{jC}^{t+1}(x_j^t, \sigma(x^t)) \\ &\quad + 2\nabla_1 f_{iB}^t(x_i^t, x_{\mathcal{N}_i^t}^t). \end{aligned}$$

However, as highlighted above, both quantities $\sigma(x^t)$ and $\sum_{j=1}^N \nabla_2 \hat{f}_{jC}^{t+1}(x_j^t, \sigma(x^t))$ are global information that, in our setting, cannot be locally accessed. In order to compensate for this lack of knowledge, we introduce two auxiliary local variables $s_i^t, y_i^t \in \mathbb{R}^3$ called trackers. As shown in (10c) and (10d), both the trackers are updated according to a perturbed consensus dynamics, [26]. Finally, in (10b) a convex combination step is performed.

Remark 4.1: The steps of Algorithm (10) resemble the ones of the scheme in [19], where a slightly different version of problem (1) without collision avoidance functions is considered. In [19], in the case of strongly convex problems, the following results are provided: (i) an upper bound for the achieved dynamic regret, and (ii) linear convergence in the static setup. Moreover, the updates x_i^{t+1} in [19], only use (old) information about f_i^t without any prediction, as we do instead in the proposed scheme.

Algorithm 1 From the perspective of robot i

initialization:

$$\begin{aligned} x_i^0 &\in X_i^0, & s_i^0 &= \phi_i(x_i^0), & y_i^0 &= \nabla_2 f_i^0(x_i^0, s_i^0) \\ \hat{\xi}_i^0 &= \text{col}(\xi_{i,p}^0, \xi_{i,b}^0) & P_i^0 &= 0_{6 \times 6} \end{aligned}$$

for $t = 0, 1, \dots$ **do**

 Measure $z_i^t = \bar{H} \xi_i^t + w_i^t$

 Predict

$$\begin{aligned} \hat{\xi}_i^{t+1} &= (\bar{F} - K_i \bar{H}) \hat{\xi}_i^t + K_i z_i^t \\ P_i^{t+1} &= (\bar{F} - K_i \bar{H}) P_i^t (\bar{F} - K_i \bar{H})^\top + K_i R_i K_i^\top + S_i \\ (\hat{f}_i^{t+1}, \hat{X}_i^{t+1}) &\leftarrow \hat{\xi}_i^{t+1} \end{aligned}$$

 Optimize

$$\tilde{x}_i^t = P_{X_i^t} [x_i^t - \alpha (\nabla_1 \hat{f}_i^t(x_i^t, s_i^t, x_{\mathcal{N}_i^t}^t) + \nabla \phi_i(x_i^t) y_i^t)] \quad (10a)$$

$$x_i^{t+1} = x_i^t + \delta (\tilde{x}_i^t - x_i^t) \quad (10b)$$

$$s_i^{t+1} = \sum_{j \in \mathcal{N}_i^t} a_{ij}^t s_j^t + \phi_i(x_i^{t+1}) - \phi_i(x_i^t) \quad (10c)$$

$$\begin{aligned} y_i^{t+1} &= \sum_{j \in \mathcal{N}_i^t} a_{ij}^t y_j^t + \nabla_2 \hat{f}_i^{t+1}(x_i^{t+1}, s_i^{t+1}, x_{\mathcal{N}_i^t}^{t+1}) \\ &\quad - \nabla_2 \hat{f}_i^t(x_i^t, s_i^t, x_{\mathcal{N}_i^t}^t) \end{aligned} \quad (10d)$$

Notice also that since the constraint set is designed according to Section III-B, then the projection step in (10a) can be performed by thresholding each component. As for the communication burden, steps (10c)-(10d) require robots to exchange at each communication round two vectors in \mathbb{R}^3 . Thereby, the exchanged data always consists of 6 floats.

V. SIMULATIVE AND EXPERIMENTAL RESULTS

In this section, we validate the proposed strategy in both a surveillance scenario (via simulations) and a basketball-

like setting (via experiments). All the simulations and experiments are performed using CHOIRBOT [27], a novel ROS 2 framework tailored for multi-robot applications. Specifically, each robot is controlled by a set of independent ROS processes implementing the distributed algorithm. To this end, processes leverage the DISROPT [28] functionalities. In ROS 2, inter-process communication is implemented via the TCP/IP stack. This allows us to implement distributed algorithms, both in simulations and experiments, on a real WiFi network. Thus, each robot only communicates with a few neighbors according to a given graph. In both the simulations and the experiment, we adopt Algorithm 1, with $\alpha = 0.2$ and $\delta = 0.4$. As for the Kalman filter, we consider $\Delta t = 0.01$ s, $S_i = 10 I_{12}$ and $R_i = 10^{-4} I_4$, for all $i \in V$.

A. Surveillance strategies

In this subsection, we consider a static scenario where a defender team of $N = 3$ robots aim to protect a (fixed) target from an intruder squad with 3 (static) members. For the sake of simplicity, we consider all the robots moving at a fixed height. We model the surveillance strategy by solving a static instance of problem (1) with costs

$$\begin{aligned} f_i^t(x_i, \sigma(x), x_{\mathcal{N}_i^t}) &= \gamma_{i,p} \|x_i - \tilde{p}_i\|^2 + \gamma_{i,\text{agg}} \|\sigma(x) - b\|^2 + \\ &\quad + \gamma_{i,b} \|\sigma(x) - x_i\|^2 + \sum_{j \in \mathcal{N}_i^t} -\log(\|x_i - x_j\|), \end{aligned}$$

for all $i \in V$, with $\gamma_{i,p}, \gamma_{i,\text{agg}} > 0$, $\tilde{p}_i \in \mathbb{R}^3$ denotes a point lying on the segment connecting the i^{th} intruder with the target, i.e., $\tilde{p}_i = \lambda_i p_i + (1 - \lambda_i) b$ with $\lambda_i \in [0, 1]$. The more the i^{th} intruder is considered dangerous, the more the defensive strategy is aggressive (i.e., $\lambda_i \rightarrow 1$). As a consequence, the i^{th} defender positions itself closer to the corresponding intruder. The impact of this parameter is shown in Figure 3. Specifically, an empirical tuning shows that, Figure 3 (left), increasing the parameters λ_i , the final configuration of the robots in V results closer to the robots in I . Instead in Figure 3 (right), we reduce the magnitude of these coefficients and the defenders place closer to the target. The parameters $\gamma_{i,b}$ are meant to enforce the defenders to keep their barycenter as close as possible to the target. The coefficients $\gamma_{i,\text{agg}}$ are used to avoid the robots to be scattered away from each other. This scenario is depicted in Figure 4. In particular, in the right picture, we boost both $\gamma_{i,b}$ and $\gamma_{i,\text{agg}}$, to highlight these effects.

B. Multi-Robot Basketball Scenario

In this subsection, we consider a multi-robot scenario related to a basketball game. The scenario concerns a team of $N = 3$ robots that play the role of the defending team, while the offending team, having also 3 players, pursues predefined trajectories. Each defending player i is associated to an offender with location $p_i^t \in \mathbb{R}^3$ at time t . Moreover, the defending team needs to take into account also the position of the ball at time t denoted by $b^t \in \mathbb{R}^3$. The strategy of each defending player i consists in placing itself as close as possible to a point lying on the segment linking the location of the basket (denoted as $p_{\text{bsk}} \in \mathbb{R}^3$) with the one of the i^{th}

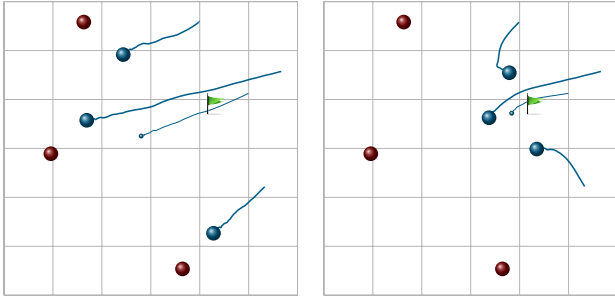


Fig. 3. Top-down view of simulation. Blue spheres denote robots in V , red spheres robots in I , green flag denotes the target. We pick $\gamma_{i,p} = 10.0$, $\gamma_{i,b} = 5.0$, $\gamma_{i,agg} = 0.1$ in both simulations, $\lambda_i = 0.8$ left, $\lambda_i = 0.2$ right.

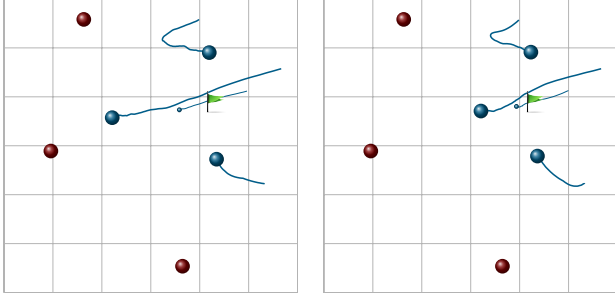


Fig. 4. Here, $\lambda_1 = 0.5$, $\lambda_2 = 0.8$, $\lambda_3 = 0.2$ and $\gamma_{i,p} = 2.0$ in both simulations, $\gamma_{i,b} = 5.0$, $\gamma_{agg} = 5.0$ (left), $\gamma_{i,b} = 20.0$, $\gamma_{agg} = 20.0$ (right)

offending player. Moreover, the defending team also aims to maintain its center of mass as close as possible to a point lying on the segment linking the basket and the ball. Finally, the robots also want to avoid collisions among them. Such a defending strategy is captured by an instance of problem (1) with objective functions

$$f_i^t(x_i, \sigma(x), x_{\mathcal{N}_i^t}) = \gamma_{i,p} \|x_i - \tilde{p}_i^t\|^2 + \gamma_{agg} \|\sigma - \tilde{b}^t\|^2 + \sum_{j \in \mathcal{N}_i^t} -\log(\|x_i - x_j\|),$$

where $\lambda_i, \lambda_{agg} \in [0, 1]$. Similarly to Section V-A, $\tilde{p}_i^t = \lambda_i p_{bsk} + (1 - \lambda_i) p_i^t$ models a point lying on the segment between the basket and the i^{th} offensive player, while $\tilde{b}^t = (1 - \lambda_{agg}) p_{bsk} + \lambda_{agg} b^t$ models a point lying on the segment between the basket and the ball. Similarly to the description in Section III-B, the feasible set of each player $i \in V$ is

$$\begin{cases} [p_i^t]_c + \epsilon_c \leq [x_i^t]_c \leq [p_{bsk}]_c & \text{if } [p_i^t]_c \leq [p_{bsk}]_c \\ [p_{bsk}]_c \leq [x_i^t]_c \leq [p_i^t]_c - \epsilon_c & \text{if } [p_i^t]_c > [p_{bsk}]_c \end{cases} \quad c = 1, 2, 3,$$

$\epsilon_c^t = \max(\epsilon_{c,\min}, \kappa_c \|[p_i^t]_c - [p_{bsk}]_c\|^2)$ with $\kappa_c, \epsilon_{c,\min} > 0$. To simulate a realistic behavior, the offending team follows predefined trajectories, and at a certain point, the ball is passed between two teammates. In the proposed experiments, we choose a robotic platform for the Crazyflie 2.0 nanoquadrotor. Intruders are virtual, simulated quadrotors. A snapshot from an experiment is in Figure 5, while a video is available as accompanying material to the paper¹.

¹The video is also available at <https://www.youtube.com/watch?v=5bFFdURhTYs>

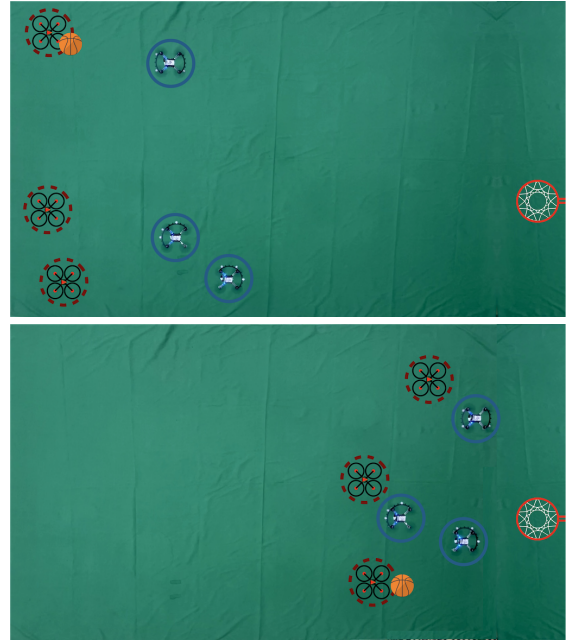


Fig. 5. Snapshot from an experiment. Real defenders are highlighted by blue circles and virtual attackers are depicted with red quadrotors.

VI. CONCLUSION

This work proposed a distributed algorithm to control a team of cooperating robots protecting a target from a set of intruders. The strategy of the defending team has been modeled according to a distributed online aggregative optimization framework. Specifically, the robotic team determines its configuration in the space according to the intruder positions, team barycenter position with respect to the target, and inter-robot collisions. We corroborated the effectiveness of our method with simulations and experiments on a team of cooperating quadrotors.

REFERENCES

- [1] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009, vol. 27.
- [2] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Intern. Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019.
- [3] A. V. Savkin and A. Marzoughi, "Distributed control of a robotic network for protection of a region from intruders," in *2017 IEEE Intern. Conf. on Robotics and Biomimetics*. IEEE, 2017, pp. 804–808.
- [4] A. Marzoughi and A. V. Savkin, "Autonomous navigation of a team of unmanned surface vehicles for intercepting intruders on a region boundary," *Sensors*, vol. 21, no. 1, p. 297, 2021.
- [5] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple uavs for persistent surveillance: Algorithm and flight test results," *IEEE TCST*, vol. 20, no. 5, pp. 1236–1251, 2011.
- [6] P. Natarajan, P. K. Atrey, and M. Kankanhalli, "Multi-camera coordination and control in surveillance systems: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 4, pp. 1–30, 2015.
- [7] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, "Cooperative coverage for surveillance of 3d structures," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1838–1845.

- [8] M. Petrлік, V. Vonásek, and M. Saska, "Coverage optimization in the cooperative surveillance task using multiple micro aerial vehicles," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 4373–4380.
- [9] E. Simetti, A. Turetta, G. Casalino, E. Storti, and M. Cresta, "Protecting assets within a civilian harbour through the use of a team of usvs: Interception of possible menaces," in *IARP workshop on robots for risky interventions and environmental surveillance-maintenance, Sheffield, UK*, 2010.
- [10] Y. Zhang and Y. Meng, "A decentralized multi-robot system for intruder detection in security defense," in *2010 IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*. IEEE, 2010, pp. 5563–5568.
- [11] E. Raboin, P. Švec, D. Nau, and S. K. Gupta, "Model-predictive target defense by team of unmanned surface vehicles operating in uncertain environments," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 3517–3522.
- [12] G. E. Mullins and S. K. Gupta, "Adversarial blocking techniques for autonomous surface vehicles using model-predictive motion goal computation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2272–2278.
- [13] V. Popov, "Robot learning from demonstration and the problem of target defense by team of unmanned surface vehicles," *Applied Mathematical Sciences*, vol. 8, no. 100, pp. 4971–4976, 2014.
- [14] J. J. Acevedo, B. Arrue, I. Maza, and A. Ollero, "A distributed framework for surveillance missions with aerial robots including dynamic assignment of the detected intruders," in *2016 Intern. Conf. on Unmanned Aircraft Systems*. IEEE, 2016, pp. 1212–1219.
- [15] S. Nantogma, W. Ran, X. Yang, and H. Xiaoqin, "Behavior-based genetic fuzzy control system for multiple usvs cooperative target protection," in *2019 3rd International Symposium on Autonomous Systems*. IEEE, 2019, pp. 181–186.
- [16] X. Duan, M. George, R. Patel, and F. Bullo, "Robotic surveillance based on the meeting time of random walks," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1356–1362, 2020.
- [17] X. Li, L. Xie, and Y. Hong, "Distributed aggregative optimization over multi-agent networks," *IEEE Trans. on Automatic Control*, 2021.
- [18] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Trans. on Contr. of Network Sys.*, 2021.
- [19] G. Carnevale, A. Camisa, and G. Notarstefano, "Distributed online aggregative optimization for dynamic multi-robot coordination," *IEEE Transactions on Automatic Control*, 2022.
- [20] Z. Chen and S. Liang, "Distributed aggregative optimization with quantized communication," *Kybernetika*, vol. 58, no. 1, 2022.
- [21] G. Carnevale, N. Mimmo, and G. Notarstefano, "Aggregative feedback optimization for distributed cooperative robotics," *IFAC-PapersOnLine*, vol. 55, no. 13, pp. 7–12, 2022.
- [22] T. Wang and P. Yi, "Distributed projection-free algorithm for constrained aggregative optimization," *arXiv:2207.11885*, 2022.
- [23] X. Li, L. Xie, and N. Li, "A survey of decentralized online learning," *arXiv:2205.00473*, 2022.
- [24] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
- [25] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.
- [26] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.
- [27] A. Testa, A. Camisa, and G. Notarstefano, "ChoiRbot: A ROS 2 toolbox for cooperative robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2714–2720, 2021.
- [28] F. Farina, A. Camisa, A. Testa, I. Notarnicola, and G. Notarstefano, "Disropt: a python framework for distributed optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2666–2671, 2020.