Distributed Feedback Optimization for Multi-Robot Target Encirclement and Patrolling

Published Version:
Pichierri, L., Carnevale, G., Notarstefano, G. (2024). Distributed Feedback Optimization for Multi-Robot Target Encirclement and Patrolling. 345 E 47TH ST, NEW YORK, NY 10017 USA : IEEE Computer Society [10.1109/case59546.2024.10711694].

Availability:
This version is available at: https://hdl.handle.net/11585/1013610 since: 2025-04-01

Published:
DOI: http://doi.org/10.1109/case59546.2024.10711694

Distributed Feedback Optimization for Multi-Robot Target Encirclement and Patrolling

Lorenzo Pichierri, Guido Carnevale, and Giuseppe Notarstefano

***Abstract*— In this paper, we address multi-robot target monitoring and patrolling tasks via distributed feedback optimization. In particular, we design a distributed policy to steer a network of peer-to-peer robots toward a configuration minimizing a comprehensive index cost taking into account three different goals. First, the robots aim at disposing in a formation enclosing a given target. Second, each single robot aims at placing itself as close as possible to a specific point of interest. Third, the robots need to avoid dangerous locations modeled according to a suitable potential. To model this overall task, we resort to the formalism of aggregative feedback optimization, a recently emerged framework in which the goal is to minimize the sum of local functions each depending on both local (e.g., the position of a robot) and global variables (e.g., the barycenter of a team of robots) while concurrently taking into account also the robots' nonlinear dynamics. We test our distributed strategy via realistic Webots simulations of a team of Crazyflie nano-quadrotors in a ROS 2 framework.**

I. Introduction

Recent years have seen an impressive increase in the employment of teams of multi-robots for a wide range of tasks, see, e.g., [1] for a comprehensive overview. In this work, we investigate the framework in which a team of robots wants to cooperatively monitor and encircle a common target while, at the same time, each robot also has to patrol a specific point of interest.

Many researchers have addressed the issue of encirclement and patrolling in multi-robot frameworks using many different approaches and methodologies. In [2], the authors propose a self-organizing algorithm for multi-robot target encirclement, considering simple robots, with limited communication and capabilities. The task of monitoring and cordoning has many links with surveillance tasks. A coverage control approach has been used in [3]–[5] for surveillance purposes. In [6], a distributed, multi-objective algorithm is designed to detect intruders and protect sensitive areas. Authors in [7], [8] address the problem with a team of autonomous vehicles aimed at maximizing the amount of time needed by an intruder to reach a target. The work [9] investigates the surveillance tasks using a team of water vehicles.

This paper models the multi-robot task by resorting to the formalism of aggregative optimization. Indeed, such an emerging framework is gaining increasing interest in the context of cooperative robotics [10], [11]. In this field, a network of robots aims to minimize an objective function given by the sum of local functions which depend on both local decision variables (e.g., the position of the robot) and an aggregation of all of them (e.g., the barycenter of the team). The first work studying this new class of problem is [12], where an unconstrained and static scenario is considered. In [13], [14], the problem has been extended to deal with constrained and time-varying scenarios. Feedback optimization in the aggregative scenario is studied in [15] (and in the related preliminary version [16]).

In fact, this latest work, unlike all the others mentioned earlier, also takes into account the dynamics of local plants whose state coincides with the decision variable of the optimization problem. Moreover, it assumes that the gradients of the unknown functions of the optimization problem can only be measured according to the current configuration. This approach is known in the literature as feedback optimization, see, e.g., [17] for a recent general survey and [18] for a distributed approach.

In this work, we propose a distributed approach for multi-robot target encirclement and patrolling. Specifically, we consider the problem of (i) cooperatively encircling a common target, while concurrently (ii) patrolling a set of specific points of interest and (iii) avoiding possible dangers modeled with a suitable potential. To accomplish these tasks in a distributed fashion, we rely on aggregative optimization, a recently emerged scenario in which the goal is to minimize the sum of local costs each depending on both network quantities (e.g., the robots' team barycenter) and local ones (e.g., single robot location). In order to handle both robot (nonlinear) control and optimization at the same stage, we consider a feedback optimization point-of-view. In particular, we adopt Aggregative Tracking Feedback [15], a recent distributed feedback optimization law for aggregative problems. This distributed algorithm combines a gradient-based controller with a consensus-based part compensating for the local lack of knowledge of the desired global quantities due to the presence of the aggregative variable in the optimization program. We show how to embed this strategy into a team of Crazyflie nano-quadrotors equipped with a stabilizing low-level controller. We test our strategy via realistic Webots-based simulations carried out in a ROS 2 frameworks. These tests show the efficacy and flexibility of the proposed distributed policy.

The paper unfolds as follows. In Section II, we introduce preliminaries about distributed feedback aggregative optimization. In Section III we detail how to address the considered tasks design with a proper distributed policy. In

This work was supported in part by the Italian Ministry of Foreign Affairs and International Cooperation, grant number BR22GR01.

Authors are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy. Email: {name.surname}@unibo.it.

Section IV, we show how to properly embed such a policy into an overall distributed startegy tailored for a team of nano-quadrotors. Finally, in Section V, we provide realistic simulations testing the proposed strategy.

Notation: $\mathrm{col}(v_1, \ldots, v_n)$ is the concatenation of the vectors $v_1, \ldots, v_n$. $\mathrm{diag}\, v$ is the diagonal matrix with the components of v on the diagonal. $\mathrm{BLKDIAG}(M_1, \ldots, M_N)$ is the matrix with $M_i \in \mathbb{R}^{n_i \times m_i}$ along the diagonal blocks. $\otimes$ is the Kronecker product. The symbol I_m is the identity matrix In $\mathbb{R}^{m \times m}$. 1_N and 0_N are the vectors of N ones and zeros, respectively, while $\mathbf{1} := 1_N \otimes I_d$. Given $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^n$, we define $\nabla_1 f(x, y) := \frac{\partial}{\partial s} f(s, y)|_{s=x}$ and $\nabla_2 f(x, y) := \frac{\partial}{\partial s} f(x, s)|_{s=y}$. Let $X \subseteq \mathbb{R}^n$ and $x \in \mathbb{R}^n$, then $\|x\|_X := \inf_{y \in X} \|x - y\|$. We omit time-dependency in every dynamical system $\dot{x}(t) = f(x(t))$.

II. Preliminaries

In this section, we recall the generic formulation of the distributed feedback aggregative optimization scenario. As it will become clearer in the next, the proposed method relies on information exchanges among the robots of the team. Thus, before introducing the whole setup, we establish the communication model connecting the cooperating robots.

A. Communication Model

In this paper, we consider a set of N robots that have to cooperatively monitor and cordone a common target. In this cooperative setting, robots have to leverage communications among the robots in order to solve the problem. We assume that robots can exchange information according to a communication network modeled as a strongly connected directed graph $\mathcal{G} = (\{1, \ldots, N\}, \mathcal{E})$ in which $\mathcal{E} \subseteq \{1, \ldots, N\} \times \{1, \ldots, N\}$ is the edge set. The graph $\mathcal{G}$ models the communication in the sense that there is an edge $(i, j), (j, i) \in \mathcal{E}$ if and only if robots i is able to exchange information with robots j. We consider $(i, i) \in \mathcal{E}$ and for each $i \in \{1, \ldots, N\}$. For each node i, the set of (in)-*neighbors* j such that there exists an edge $(i, j) \in \mathcal{E}$ of i at time t is denoted by $\mathcal{N}_i = \{j \in \{1, \ldots, N\} \mid (i, j) \in \mathcal{E}\}$. We also associate a so-called weighted adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ matching the graph, i.e., a matrix with (i, j)-entry $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$, otherwise $a_{ij} = 0$. The weighted in-degree and out-degree of agent i are defined as $deg_i^{\text{in}} = \sum_{j \in \mathcal{N}_i} a_{ij}$ and $deg_i^{\text{out}} = \sum_{j \in \mathcal{N}_i} a_{ji}$, respectively. In this paper, the graph $\mathcal{G}$ is weight-balanced, namely $deg_i^{\text{in}} = deg_i^{\text{out}}$ for all $i \in \{1, \ldots, N\}$. Finally, we associate to $\mathcal{G}$ the so-called Laplacian matrix defined as $\mathcal{L} := \mathcal{D}^{\text{in}} - \mathcal{A}$, where $\mathcal{D}^{\text{in}} := \mathrm{diag}(deg_1^{\text{in}}, \ldots, deg_N^{\text{in}}) \in \mathbb{R}^{N \times N}$.

B. Distributed Feedback Aggregative Optimization

In this framework, there are N robots each evolving according to the nonlinear generic dynamics

$$\dot{x}_i = r_i(x_i, u_i), \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$ are the state and the input of robot i, respectively, and $r_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ represents its dynamics. As customary in the field of feedback optimization, the dynamics described in (1) is already stabilized, i.e., the physical plant of robot i is already endowed with a local low-level controller embedded into the map r_i. More formally, for all $i \in \{1, \ldots, N\}$, this property ensures the existence of a steady-state map $h_i : \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ such that, for any (fixed) control input $\bar{u}_i \in \mathbb{R}^{m_i}$, the point $h_i(\bar{u}_i)$ is a globally exponentially stable equilibrium for robot i (cf. (1)) In other words, in this setting, each input u_i can be seen as the reference signal provided to (1) representing the closed-loop system of robot i. In particular, the goal is to design a control law $u := \mathrm{col}(u_1, \ldots, u_N) \in \mathbb{R}^m$, with $m := \sum_{i=1}^N m_i$, able to steer the robots to a configuration being a solution of the aggregative optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n, u \in \mathbb{R}^m} \quad & \sum_{i=1}^N f_i(x_i, \sigma(x)). \\ \text{subj. to } \; & x_i = h_i(u_i), \forall i \in \{1, \ldots, N\}, \end{aligned} \tag{2}$$

where $x = \mathrm{col}(x_1, \ldots, x_N) \in \mathbb{R}^n$, with $n := \sum_{i=1}^N n_i$, and each cost $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^a \mapsto \mathbb{R}$ depends on both the local decision variable of robot i and on an *aggregative variable* $\sigma : \mathbb{R}^n \to \mathbb{R}^a$ given by

$$\sigma(x) := \frac{1}{N} \sum_{i=1}^N \phi_i(x_i),$$

with $\phi_i(x_i) : \mathbb{R}^{n_i} \mapsto \mathbb{R}^a$ for each $i \in \{1, \ldots, N\}$. We also introduce the overall objective function $f^\sigma : \mathbb{R}^n \to \mathbb{R}$ and its reduced version $f^{\sigma,h} : \mathbb{R}^m \to \mathbb{R}$ defined as

$$f^\sigma(x) := \sum_{i=1}^N f_i(x_i, \sigma(x)) \tag{3a}$$

$$f^{\sigma,h}(u) := \sum_{i=1}^N f_i(h_i(u_i), \sigma(h(u))). \tag{3b}$$

Throughout this paper, we consider a cooperative, distributed scenario in which robots have limited knowledge of the problem (2). Thus, in order to solve the optimization problem, they have to iteratively exchange suitable information via a communication network. The analytic expression of the local costs f_i and aggregation rules ϕ_i are not available to the robots, they can be only measured according to current local variables. Specifically, each robot i can only access $\nabla_1 f_i(x_i, \hat{\sigma}_i)$, $\nabla_2 f_i(x_i, \hat{\sigma}_i)$, $\phi_i(x_i)$, $\nabla \phi_i(x_i)$, and $\nabla h_i(u_i)$ where x_i and u_i are its current state and input variables, respectively, while $\hat{\sigma}_i \in \mathbb{R}^a$ is its local estimate of $\sigma(x)$.

C. Aggregative Tracking Feedback

In order to steer the robots modeled in (1) toward a configuration representing a solution of the time-varying problem (2), we adopt Aggregative Tracking Feedback, i.e., the distributed feedback optimization law proposed in [15]. The strategy is derived by considering the so-called reduced version of problem (2), i.e., the one obtained by considering each robot i already in the steady-state configuration $x_i = h_i(u_i)$. Such a problem reads as

$$\min_{u\in\mathbb{R}^m} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u))). \tag{4}$$

Then, by deriving the whole reduced cost $f^{\sigma,h}(u)$ (cf. (3b)) with respect to u_i, each robot i should implement the gradient-flow controller

$$\begin{aligned}\dot{u}_i = &-\nabla h_i(u_i)\Big(\nabla_1 f_i(h_i(u_i), \sigma(h(u))) \\ &+ \frac{\nabla\phi_i(h_i(u_i))}{N}\sum_{j=1}^{N}\nabla_2 f_{j,t}(h_j(u_j), \sigma(h(u)))\Big),\end{aligned} \tag{5}$$

where $u := \text{col}(u_1, \dots, u_N) \in \mathbb{R}^m$. First, keeping in mind the desired distributed paradigm of our strategy, the global quantity $\sigma(x)$ and $\sum_{j=1}^{N}\nabla_2 f_j(x_j, \sigma(x))$ are not available for robot i. Second, as customary in the context of feedback optimization, robot i can only measure the problem quantities $\nabla_1 f_i(x_i, \hat{\sigma}_i)$, $\nabla\phi_i(x_i)$, and $\sum_{j=1}^{N}\nabla_2 f_{j,t}(x_j, \hat{\sigma}_j)$, where $\hat{\sigma}_i \in \mathbb{R}^a$ is the local estimate of agent i about $\sigma(x)$, where $x := \text{col}(x_1, \dots, x_N) \in \mathbb{R}^n$. Therefore, dynamics (5) is modified by using the available measures and introducing two auxiliary variables $w_i, z_i \in \mathbb{R}^a$ compensating the lack of knowledge about $\sigma(x)$ and $\sum_{j=1}^{N}\nabla_2 f_{j,t}(x_j, \hat{\sigma}_j)$ via a continuous-time tracking mechanism. The overall strategy from the perspective of robot i reads as

$$\dot{u}_i = -\alpha_1 d_i(x_i, u_i, w_i, z_i) \tag{6a}$$

$$\dot{w}_i = -\frac{1}{\alpha_2}\sum_{j\in\mathcal{N}_i} a_{ij}\left(w_i + \phi_i(x_i) - w_j - \phi_j(x_j)\right) \tag{6b}$$

$$\begin{aligned}\dot{z}_i = &-\frac{1}{\alpha_2}\sum_{j\in\mathcal{N}_i} a_{ij}(z_i + \nabla_2 f_i(x_i, w_i + \phi_i(x_i))) \\ &+ \frac{1}{\alpha_2}\sum_{j\in\mathcal{N}_i} a_{ij}(z_j + \nabla_2 f_j(x_j, w_j + \phi_j(x_j))),\end{aligned} \tag{6c}$$

where $d_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \times \mathbb{R}^a \times \mathbb{R}^a \to \mathbb{R}^{m_i}$ reads as

$$\begin{aligned}&d_i(x_i, u_i, w_i, z_i) \\ &:= \nabla h_i(u_i)\nabla_1 f_i(x_i, w_i + \phi_i(x_i)) \\ &\quad + \nabla h_i(u_i)\nabla\phi_i(x_i)\left(\nabla_2 f_i(x_i, w_i + \phi_i(x_i)) + z_i\right),\end{aligned}$$

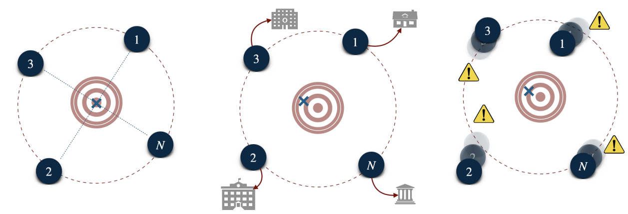
for all $i \in \{1, \dots, N\}$. The parameter $\alpha_1 > 0$ tunes the speed of u_i and (w_i, z_i) relative to the plant. In turn, the parameter $\alpha_2 > 0$ tunes the speed of (w_i, z_i) relative to u_i. In order to implement dynamics (6), each robot i needs to exchange variables only with its neighbors and, thus, the proposed solution is purely distributed. More in detail, the dynamics (6b)-(6c) require robots to exchange two vectors in $\mathbb{R}^a$. Thereby, the exchanged data consists of $2d$ floats. In a forthcoming journal version of the work, we will also consider event-triggered communication in place of the unrealistic continuous-time one considered here.

III. Distributed Strategy for Multi-Robot Target Encirclement and Patrolling

In this section, we design a distributed strategy to address multi-robot target encirclement and patrolling. More in detail, the goal of the distributed strategy is to steer each robot state x_i in a configuration chosen according to the following policies

1) cooperatively target encirclement: to force the formation in a configuration enclosing a given target;
2) multi-spots patrolling: to place itself as close as possible to a specific point of interest;
3) safe placement: to take into account a given potential modeling the (possible) presence of dangers.

An illustrative example of this setup is provided in Fig. 1.



(a) Placement 1 (b) Placement 2 (c) Placement 3

Fig. 1: Illustrative example of desired placement. Each robot i wants to move toward a specific spot and avoids dangerous configurations, while the cooperative target encirclement turns out as an emerging behavior of the team.

To this end, we rely on the formalism of time-varying distributed feedback aggregative optimization (see Section II). In particular, we consider an instance of problem (2) in which we model the local cost function $f_i(x_i, \sigma(x))$ as the sum of three terms, namely

$$f_i(x_i, \sigma(x)) = f_i^{\text{enc}}(\sigma(x)) + f_i^{\text{spot}}(x_i) + f_i^{\text{danger}}(x_i),$$

where

1) $f_i^{\text{enc}} : \mathbb{R}^a \to \mathbb{R}$ models task 1), i.e., the cooperative target encirclement;
2) $f_i^{\text{spot}} : \mathbb{R}^{n_i} \to \mathbb{R}$ models task 2), i.e., the multi-spots patrolling;
3) $f_i^{\text{danger}} : \mathbb{R}^{n_i} \to \mathbb{R}$ models task 3), i.e., the need for placing in a safe configuration with respect to a considered potential.

In order to define these cost functions, we need to introduce $\mathrm{p}_i(x_i) \in \mathbb{R}^2$ as the cartesian planar position of robot i and $\mathrm{q}_i(x_i) \in \mathbb{R}^2$ as the polar coordinates with respect to the target $b := \text{col}(b_1, b_2) \in \mathbb{R}^2$, namely

$$\mathrm{p}_i(x_i) := \begin{bmatrix}\mathrm{p}_{1,i}(x_i)\\ \mathrm{p}_{2,i}(x_i)\end{bmatrix}, \quad \mathrm{q}_i(x_i) := \begin{bmatrix}\rho_i(x_i)\\ \theta_i(x_i)\end{bmatrix}.$$

With this notation at hand, we define the *aggregative variable* to contain two contributions: (i) the cartesian barycenter of the team and (ii) the mean angle of the polar coordinates

$\theta_i(x_i)$ of robot i, namely $\sigma : \mathbb{R}^n \to \mathbb{R}^3$ reads as

$$\sigma(x) = \begin{bmatrix} \sigma_{\mathrm{p}}(x) \\ \sigma_\theta(x) \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{N} \mathrm{p}_i(x_i) \\ \frac{1}{N}\sum_{i=1}^{N} \theta_i(x_i) \end{bmatrix}.$$

In the following, we detail the modeling of the cost terms $f_i^{\text{enc}}(\sigma(x))$, $f_i^{\text{spot}}(x_i)$, and $f_i^{\text{danger}}(x_i)$.

A. Cooperative Target Encirclement

The cooperative target encirclement is forced via the cost function $f_i^{\text{enc}}(\sigma(x))$ which, in turn, is given by the sum of three terms, namely

$$f_i^{\text{enc}}(\sigma(x)) = f_i^{\text{enc},1}(\sigma(x)) + f_i^{\text{enc},2}(\sigma(x)) + f_i^{\text{enc},3}(\sigma(x)),$$

with

$$\begin{aligned} f_i^{\text{enc},1}(\sigma(x)) &= \gamma_1 \left\| \sigma_{\mathrm{p}}(x) - b \right\|^2 \\ f_i^{\text{enc},2}(\sigma(x)) &= \gamma_2 \left\| \sigma_\theta(x) - \mu_\theta \right\|^2 \\ f_i^{\text{enc},3}(\sigma(x)) &= -\gamma_3 \left\| \sigma_\theta(x) - \theta_i \right\|^2, \end{aligned}$$

where $\gamma_1, \gamma_2, \gamma_3 > 0$ are tuning parameters and μ_θ the desired mean angle of the polar coordinates of the robots' team. In particular, the first term $f_i^{\text{enc},1}(\cdot)$ aims at steering the robots toward the target b. As for the second term, it forces the robot toward configurations having the desired mean angle μ_θ. Finally, the third term aims at maximizing the variance of the angular position of the robots, spreading them in the interval $[0, 2\pi]$ and, thus, brings to a more uniform distribution of the robots around the target.

B. Cooperative Encirclement of a Common target with Specific Targets Protection

We now design the cost term $f_i^{\text{spot}}(x_i)$ to embed in the strategy also the neeed for monitoring some specific locations. More in detail, given $S \in \mathbb{N}$ point of interest, we force each robot i to stay as close as possible to a point s_ℓ with $\ell \in \{1, \dots, S\}$ by defining $f_i^{\text{spot}}(x_i)$ as

$$f_i^{\text{spot}}(x_i) = \gamma_4 \left\| \mathrm{p}_i(x_i) - s_\ell \right\|^2,$$

where $\gamma_4 > 0$ is a tuning parameter and $s_\ell \in \mathbb{R}^2$ the location of the ℓ-th point of interest, described in polar coordinates with respect to the target position b.

C. Safe-Positioning

We model the need for robots' safe-positioning by introducing a potential $U_i : \mathbb{R}^{n_i} \to \mathbb{R}$ penalizing the configurations close to dangers with higher costs, namely

$$f_i^{\text{danger}}(x_i) = \gamma_5\, U_i(x_i),$$

for all $i \in \{1, \dots, N\}$, where $\gamma_5 > 0$ is a tuning parameter.

IV. Distributed Strategy Structure for a Team of Crazyflies

In this section, we show how to design a distributed strategy embedding Aggregative Tracking Feedback to address cooperative multi-robot tasks over a team of N Crazyflies. The model of each Crazyflie i including also a stabilizing low-level PID Attitude Rate controller, can be written as

$$\dot{p}_i = v_i \tag{7a}$$

$$\dot{v}_i = g z_W - \frac{\xi_{1,i}}{m} R(\eta_i) z_W - \frac{1}{m} R(\eta_i) A R(\eta_i)^\top v_i \tag{7b}$$

$$\dot{\eta}_i = \begin{bmatrix} \xi_{2,i} \\ \xi_{3,i} \\ \xi_{4,i} \end{bmatrix}, \tag{7c}$$

where $p_i \in \mathbb{R}^3$ is the position of the nano-quadrotor in the world frame $\mathcal{W} = \{x_W, y_W, z_W\}$, $v_i \in \mathbb{R}^3$ is its linear velocity, while the vector $\eta_i \in \mathbb{R}^3$ collects the roll, pitch, and yaw angles denoted by $\varphi_i, \vartheta_i, \psi_i$. Hence, the whole robot state is $x_i := \begin{bmatrix} p_i^\top & v_i^\top & \eta_i^\top \end{bmatrix}^\top \in \mathbb{R}^9$ for all $i \in \{1, \dots, N\}$. In (7), $m = 0.027$ Kg is the mass, and g is the gravity coefficient. The rotation matrix is denoted by $R \in SO(3)$, and The term $RAR^\top v_i$ in (7b) refers to the rotor drag effect as modeled in [19] and has been neglected. The symbol $\xi_i = [\xi_{1,i}\ \xi_{2,i}\ \xi_{3,i}\ \xi_{4,i}]^\top \in \mathbb{R}^4$ is used to denote the thrust and angular rate commands used for the low-level controller of the Crazyflie. In particular, $\xi_{1,i} \in \mathbb{R}$ denotes the thrust and $\xi_{2,i}, \xi_{3,i}, \xi_{4,i} \in \mathbb{R}$ the angular rates (according to the $Z-Y-X$ extrinsic Euler representation) generated according to the nonlinear geometric controller proposed by [20] and fed by the state x_i and the variable $u_i \in \mathbb{R}^2$ provided by Aggregative Tracking Feedback. Hence, we model the generation of each command ξ_i as

$$\xi_i = g(x_i, u_i), \tag{8}$$

in which $g : \mathbb{R}^9 \times \mathbb{R}^2 \to \mathbb{R}^4$ describes the nonlinear geometric controller proposed by [20]. It is possible to show the stabilizing properties of the cascade between the geometric controller [20] and the PID Attitude Rate controller. Thus, in this scenario, the steady-state map $h_i : \mathbb{R}^2 \to \mathbb{R}^9$ reads as

$$h_i(u_i) = \begin{bmatrix} u_i^\top & z_i^{\text{static}} & 0_3^\top & 0_3^\top \end{bmatrix}^\top, \tag{9}$$

for all $i \in \{1, \dots, N\}$, where $z_i^{\text{static}} \in \mathbb{R}$ is the (static) desidered height set-point of nano-quadrotor i. Therefore, the generic robot dynamics formulation (1) is recovered by setting the robot model r_i as the composition between (7) and the low-level PID (8). As we better detail in Section V, we use CrazyChoir ROS 2 Toolbox [21] to simulate the behavior of the team of Crazyflies. The block diagram describing the overall strategy from the perspective of the nano-quadrotor i is depicted in Figure 2.

V. Simulative Results

In this section, we show the effectiveness of our distributed strategy over an applicative scenario considering a team of $N = 6$ Crazyflie nano-quadrotors addressing the multi-robot task described in Section III formulated as an

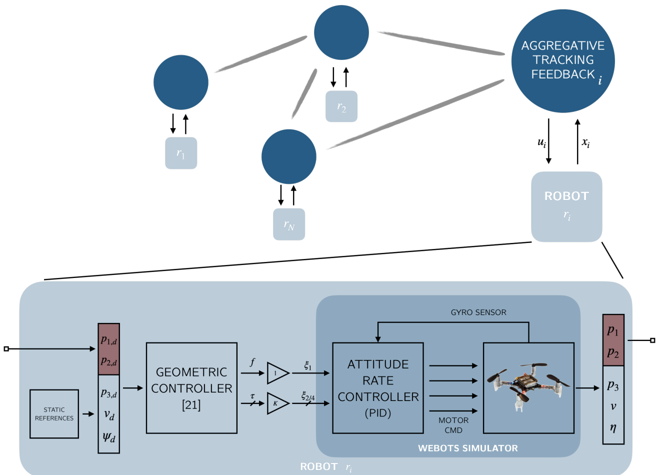


Fig. 2: Block diagram scheme illustrating the plant considered in the simulative scenario. The dynamics of the Crazyflie and the low-level PID controller are provided by the simulator engine in CrazyChoir. Moreover, the toolbox provides also the nonlinear geometric controller that governs the nano-quadrotor.

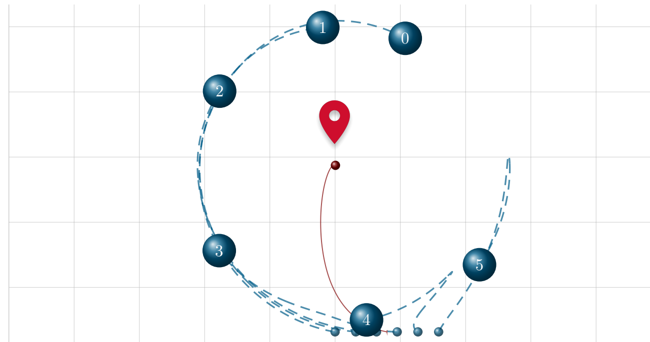


Fig. 3: Top-down view of the Scenario 1 simulation. The common target is represented by a red pinpoint image. The starting positions of the nano-quadrotors are depicted as small blue spheres, the final ones by bigger, numbered, spheres. Crazyflies' trajectories are represented by blue dashed lines, whereas their barycenter with a red line.

instance of problem (2). These simulations are performed using CrazyChoir [21], a novel ROS 2 framework tailored for Crazyflies applications based on ChoiRbot [22], another ROS 2 framework for multi-robot. In particular, each Crazyflie is governed by an independent ROS 2 control stack and supervised by a decentralized guidance layer that implements the distributed algorithm. To this end, DISROPT [23] functionalities are implemented to run the distributed algorithm among the set of nano-quadrotors. In fact, each Crazyflie communicates with a few neighbors according to a given strongly connected and weight-balanced graph. Moreover, the CrazyChoir supports simulations of the nano-quadrotors in a virtual environment, which is based on the Webots simulator. In the following, we will incrementally show the effects of the different cost terms designed in Section III. As for their tuning parameters, we choose $\gamma_1 = 3$, $\gamma_2 = 0.1$, $\gamma_3 = 0.01$, $\gamma_4 = 2$, and $\gamma_5 = 0.016$. We empirically set the tuning parameters of Aggregative Tracking Feedback according to $\alpha_1 = 0.08$ and $\alpha_2 = 0.001$. We randomly generate an Erdős-Rényi graph with edge probability 0.5.

A. Scenario 1

Here, we test our distributed strategy by considering only the cooperative encirclement task, namely

$$f_i(x_i, \sigma(x)) = f_i^{\text{enc}}(\sigma(x)),$$

for all $i \in \{1, \dots, N\}$. In Figure 3, we graphically show the effect of $f_i^{\text{enc}}(\sigma(x))$ by providing the top-down view of a simulation, conducted in a Webots environment, in which only this term has been considered. Here, we set $\mu_\theta = 0$ and $b = \text{col}(0, -1)$. A team of $N = 6$ Crazyflies starts in line and after the hovering phase, they cooperatively encircle the common wildfire area without the help of a central unity.

We consider the Crazyflies hovering at different heights, in order to avoid collisions, i.e., $z_i^{\text{static}} \neq z_j^{\text{static}}$ for all $i \neq j$.

B. Scenario 2

In this second scenario, we also consider the term $f_i^{\text{spot}}(x_i)$ to embed the need for multi-spots patrolling, namely

$$f_i(x_i, \sigma(x)) = f_i^{\text{enc}}(\sigma(x)) + f_i^{\text{spot}}(x_i).$$

In this scenario, we suppose to have $S = 6$ points of interest to protect, and we assign to each nano-quadrotor a specific point to monitor. The planar coordinates s_ℓ are randomly generated by extracting them with a uniform probability from the set $[-7, 7] \times [-7, 7]$. Clearly, more nano-quadrotors can be assigned to the same point of interest. In Figure 4 we show the top-down view of the considered simulation. The figure shows that their sequence of arrival is different from the one of the previous scenario since they now patrol the points of interest while concurrently encircling the common target.

C. Scenario 3

In this final scenario, we also consider the presence of dangers, namely we include the term $f_i^{\text{danger}}(x_i)$ in the *local* cost function $f_i(x_i, \sigma(x))$, thus obtaining

$$f_i(x_i, \sigma(x)) = f_i^{\text{enc}}(\sigma(x)) + f_i^{\text{spot}}(x_i) + f_i^{\text{danger}}(x_i).$$

In particular, we consider the presence of $D = 3$ dangers, each with planar location $d_\ell \in \mathbb{R}^2$, and the cost

$$f_i^{\text{danger}}(x_i) = \gamma_5 \sum_{\ell=1}^{D} \frac{1}{\|\text{p}_i(x_i) - d_\ell\|^2},$$

where $\gamma_5 > 0$ is a tuning parameter. In Figure 5 we show the results of a final simulation in which the nano-quadrotors change their positioning based on the dangers' locations. The latter have been randomly generated extracting them with a uniform probability from the set $[-7, 7] \times [-7, 7]$. We show that the robots cooperatively encircle the common target, position themselves near the points of interest, and avoid dangers' locations.

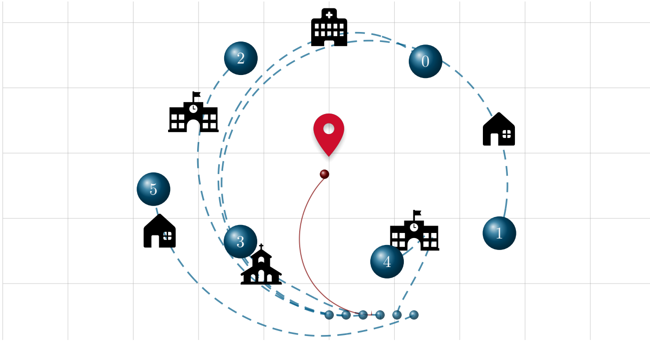


Fig. 4: Top-down view of the Scenario 2 simulation. The starting position of the Crazyflies is still the same, denoted by little blue spheres. Similarly, the common target is represented by a red pinpoint image. The points of interest are depicted by building images (schools, hospitals, etc.). The final position of the team is represented by bigger, numbered, spheres. Crazyflies' trajectories are represented by blue dashed lines.

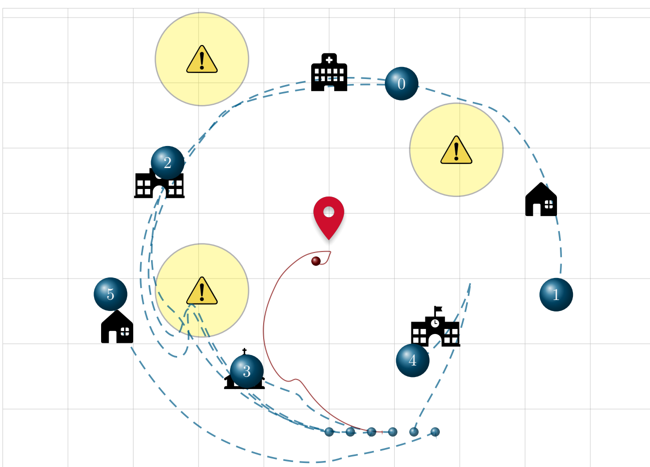


Fig. 5: Top-down view of Scenario 3. Here, the dangers' locations are depicted by yellow circles.

VI. Conclusion

This paper employed distributed feedback optimization for target monitoring and patrolling tasks arising in cooperative robotics. We taken into account three different goals. First, we considered the coooperative encirclement of a given target. Second, multi-spot monitoring has been included, while the third goal forced the avoidance of possible dangers modeled through a proper potential. We designed a distributed strategy able to steer a network of robots without a central unit toward a configuration minimizing a cost modeling the above tasks. In particular, we build an aggregative problem and, to take into account also the robots' nonlinear dynamics, we addressed it according to the feedback optimization paradigm. Realistic Webots simulations of a team of Crazyflie nano-quadrotors in a ROS 2 framework confirmed the effectiveness of our distributed policy.

References

[1] 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.

[2] A. M. Deshpande, R. Kumar, M. Radmanesh, N. Veerabhadrappa, M. Kumar, and A. A. Minai, "Self-organized circle formation around an unknown target by a multi-robot swarm using a local communication strategy," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4409–4413.

[3] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple uavs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2011.

[4] 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.

[5] M. Petrlík, 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] L. Pichierri, G. Carnevale, L. Sforni, A. Testa, and G. Notarstefano, "A distributed online optimization strategy for cooperative robotic surveillance," pp. 5537–5543, 2023.

[11] A. Testa, G. Carnevale, and G. Notarstefano, "A tutorial on distributed optimization for cooperative robotics: from setups and algorithms to toolboxes and research directions," *arXiv preprint arXiv:2309.04257*, 2023.

[12] X. Li, L. Xie, and Y. Hong, "Distributed aggregative optimization over multi-agent networks," *IEEE Trans. on Automatic Control*, 2021.

[13] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Trans. on Contr. of Network Sys.*, 2021.

[14] G. Carnevale, A. Camisa, and G. Notarstefano, "Distributed online aggregative optimization for dynamic multi-robot coordination," *IEEE Transactions on Automatic Control*, 2022.

[15] G. Carnevale, N. Mimmo, and G. Notarstefano, "Nonconvex distributed feedback optimization for aggregative cooperative robotics," *Automatica*, vol. 167, p. 111767, 2024.

[16] ——, "Aggregative feedback optimization for distributed cooperative robotics," *IFAC-PapersOnLine*, vol. 55, no. 13, pp. 7–12, 2022.

[17] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," *Annual Reviews in Control*, vol. 57, p. 100941, 2024.

[18] A. Terpin, S. Fricker, M. Perez, M. H. de Badyn, and F. Dörfler, "Distributed feedback optimisation for robotic coordination," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3710–3715.

[19] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189–8195, 2017.

[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[21] L. Pichierri, A. Testa, and G. Notarstefano, "Crazychoir: Flying swarms of crazyflie quadrotors in ros 2," *IEEE Robotics and Automation Letters*, 2023.

[22] 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.

[23] 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.