This is the author's version of an article published in IEEE Robotics and Automation Letters.
The publisher's version and the citation information are available at https://doi.org/10.1109/LRA.2024.3396643

Affine Multirobot Formation Control Based on a Modular Team Structure

Miguel Aranda and Alejandro Perez-Yus

***Abstract*—We present a leaderless distributed controller for driving a multirobot team in a planar workspace toward an affine formation, i.e., an affine transformation of a nominal configuration. Our central idea is to organize the team in interlaced modules. Specifically, we define a module as a group of four robots, and we consider two given modules interlaced if they have three robots in common. For each module we define a cost, based on least-squares affine alignment between the positions of the four robots in the current and nominal configurations. Our strategy for formation achievement is to make the robots descend along the gradient of the sum of module costs. Based on this strategy, we propose a distributed control law considering the single-integrator dynamic model. Our main contribution is that the proposed modular approach allows design and reconfiguration to be done locally, i.e., involving only the robots that belong to the modules being designed or reconfigured. We present a formal stability study and an implementation algorithm. To motivate the practical interest of the proposed approach, we illustrate its usage in a multitarget enclosing and tracking scenario. The approach is experimentally validated using simulations and tests with physical unicycle robots.**

***Index Terms*—Autonomous agents, distributed robot systems, multi-robot systems.**

I. Introduction

FORMATIONS are of interest in numerous scenarios where a multirobot task requires, or benefits from, deploying the robots according to a desired orderly arrangement [1]–[3]. These scenarios include target tracking [4], [5], collective navigation [6]–[8] and transport of objects [9]–[11]. In this article, we study affine formations, which are a type of formation actively explored in recent times [6], [12]–[21] starting from formulations presented in [6], [12]. Affine formations allow achieving a multirobot team arrangement equal to a nominal configuration up to any combination of translation, rotation, scaling, and shearing. Therefore, they are more flexible than, e.g., translational [22], [23], rigid [1], [24], [25] or shape-preserving [2], [26], [27] formations; hence, they make the team more capable of adapting its shape to the needs of a given task.

The study [12] laid the foundations for distributed affine formation control based on consensus-like algorithms. In [13], affine team maneuvers were achieved via purposely designed modifications of control weights. An approach enabling the combined control of translational, shape-preserving and affine formations was presented in [14]. The controllers in [12]–[14] were leaderless, as was the affine controller proposed in [15], where additional terms were used to also achieve translational and rigid formations with the guidance of a set of leader robots. The work [6] presented a setup where leader robots dictate maneuvers for an affine formation, and provided control algorithms for different dynamic models. A number of later works built upon the formulation of [6]. For instance, the study [16] considered high-order dynamics and studied the coordination between leaders. The authors of [17] designed controllers for high-order dynamics, using directed graphs to model the robots' interactions. In [18], directed graphs were assumed too, in a multi-layer control approach for robots with Euler-Lagrange dynamics. Such dynamics were also assumed in [19] with an observer-controller type of approach. The controller in [20] was based on the estimation of planar formation parameters dictated by a single leader. In [21], the focus was on achieving prescribed-time control convergence. Most of these affine formation controllers assume the graph is undirected, and have as a core element a *stress matrix* which is computed at the design stage in a centralized manner. The robots then use the coefficients of the matrix as the weights of their distributed control algorithm. Some formulations, e.g., in [12], [18], avoid such centralized computation. In [6], [12]–[21], the conditions for convergence to an affine formation are given as global properties of the formation's nominal configuration, the stress matrix, and/or the interaction graph.

This article takes a different perspective compared to the studies above, proposing a novel modular approach for affine formation control. We focus our attention on planar formations, which are of widespread interest in many tasks. Our approach is based on organizing the team in interlaced modules. A module is a group of four robots such that each can measure the relative state of the others in the group. Two modules are considered interlaced if they share three robots. We define a cost for each module based on least-squares affine alignment between the four robots' positions in the current and nominal configurations, and we define a team cost as the sum of module costs. Then, we show that, if any two modules can be linked by a sequence of consecutive pairwise interlacings, driving the team cost to zero implies achieving an

Manuscript received: November 29, 2023; Revised: March 3, 2024; Accepted: April 15, 2024.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by MCIN/AEI/10.13039/501100011033, the ERDF A way of making Europe, and the European Union NextGenerationEU/PRTR under Projects PID2021-124137OB-I00, PID2021-125209OB-I00, and TED2021-130224B-I00. The work of Miguel Aranda was supported by the Spanish Ministry of Universities and the European Union-NextGenerationEU under a María Zambrano Fellowship.

The authors are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain miguel.aranda@unizar.es, alperez@unizar.es

Digital Object Identifier 10.1109/LRA.2024.3396643

affine formation. Exploiting this fact, we propose a control law for single-integrator robot dynamics based on gradient descent on the team cost. We prove the convergence of the team to an affine formation under this control law. We also present an implementation algorithm, and validate the performance of our approach using simulations and experiments with physical robots having unicycle kinematics.

The prior study most closely related to ours is [27], which dealt with shape-preserving formations (i.e., translations, rotations and uniform scalings of a nominal configuration), and considered a team organized in three-robot modules (i.e., triads). Here, we exploit a similar strategy to the one proposed in [27], applied on a different formation type (affine). This requires substantial modifications: modules (which are now four-robot), shape alignment costs, and conditions for interlacing all differ from those considered in [27]. We use four robots per module instead of three due to the higher number of degrees of freedom of an affine transformation compared to a shape-preserving one. While, to our knowledge, graphs formed by four-robot modules have not been used previously in affine formation control, other works [25], [28] used structures in modules, i.e., graph cliques, for rigid formations (i.e., translations and rotations of a nominal configuration), a different type of formation from the one we consider here. The study [29] addressed the problem of achieving resilient consensus in a robot swarm, grouping robots in modules that traversed closed paths. Finally, the work [30] used affine shape alignment of clusters of points to model a deformable object for computer graphics applications. In this article, we use a similar shape alignment strategy in our affine formation controller.

With the approach we propose, the final configuration reached by the team is static and depends on the initial positions of the robots. Differently from a significant number of existing works where leader robots are used [6], [15]–[21], our approach is leaderless. This makes the system more adaptable to external task constraints, as the team's configuration is not dictated by leader robots. Exploiting this adaptability, we present a strategy that uses our approach for target enclosing [4], [5], [31], which can have applications in, e.g., multitarget-multisensor perception [32]–[34].

Contribution. We detail our contribution next. Compared to previous work on affine formation control [6], [12]–[21], our novel control approach is based on local shape alignment at the level of each four-robot group (i.e., module) and local interlacing of the modules, instead of on global properties of the system. This modularity of our approach facilitates an incremental design and reconfiguration of the system, which can be done locally, i.e., involving only the robots in the modules being designed or reconfigured. The approach we propose does not need any centralized computation at the design stage, unlike most existing approaches. Finally, we also illustrate the use of our affine formation controller in a higher-level task of multitarget enclosing and tracking.

A. Notation

The symbols $\|\cdot\|$, $\otimes$, $\mathbf{I}_n$ and $\mathbf{1}_n$ denote, respectively, the Euclidean norm (i.e., Frobenius norm for matrices), the Kronecker product, the $n \times n$ identity matrix, and a column vector of n ones. To translate a set of n points so that their centroid is zero, we will use a symmetric centering matrix denoted by $\mathbf{K}_n = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^\intercal \in \mathbb{R}^{n\times n}$. We denote an unordered set by $\{\cdot\}$, and an ordered sequence by $(\cdot)$.

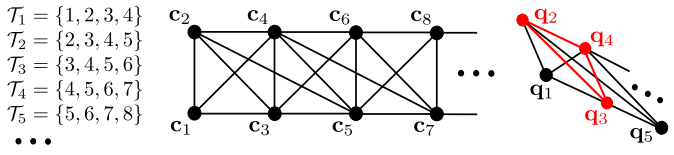

Fig. 1. Left and center: Example of graph structure based on chained modules and nominal configuration satisfying Assumptions 1 and 2: the modules $\mathcal{T}_l$ and nominal positions $\mathbf{c}_i$ with the graph edges of $\mathcal{G}$ as lines are shown. Right: example of positions $\mathbf{q}_i$ where $\mathcal{T}_1$ and $\mathcal{T}_2$ are each in a four-robot affine formation with respect to $\mathbf{c}_i$. As the positions of the three robots they share (2, 3, 4, in red) determine uniquely the affine transformation for each individual module, this transformation is the same for both.

II. Problem Formulation

We consider a team of n robots, where $n > 3$, and assign each of them a different index in the set $\mathcal{N} = \{1, \dots, n\}$.

Robot model. For each robot i, we define its current position as a point $\mathbf{q}_i \in \mathbb{R}^2$, expressed in an arbitrary fixed reference frame. A stack vector of these positions for the full team is then denoted by $\mathbf{q} = [\mathbf{q}_1^\intercal, \dots, \mathbf{q}_n^\intercal]^\intercal \in \mathbb{R}^{2n}$. We assume that the dynamic model of the robots is single-integrator, i.e., $\dot{\mathbf{q}}_i = \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{R}^2$ is the control input of robot i and $\mathbf{u} = [\mathbf{u}_1^\intercal, \dots, \mathbf{u}_n^\intercal]^\intercal \in \mathbb{R}^{2n}$ is the team control input such that $\dot{\mathbf{q}} = \mathbf{u}$. Further details regarding our requirements on the robots in terms of sensing, reference frames, and communications are given in Sec. III-C.

Nominal configuration and affine formation. We consider a *nominal configuration* [6] defined as a set of n positions, $\mathbf{c}_i \in \mathbb{R}^2$ for each robot $i \in \mathcal{N}$, stacked in $\mathbf{c} = [\mathbf{c}_1^\intercal, \dots, \mathbf{c}_n^\intercal]^\intercal \in \mathbb{R}^{2n}$. The nominal configuration is defined as constant.

Definition 1 (Affine formation): The team is in an affine formation if $\mathbf{q} \in \mathcal{S}$, where $\mathcal{S}$ is the affine image of $\mathbf{c}$ [12], defined by any of the following three equivalent expressions:

$$\mathcal{S} = \{\mathbf{q} \in \mathbb{R}^{2n} : \mathbf{q} = (\mathbf{I}_n \otimes \mathbf{G})\mathbf{c} + \mathbf{1}_n \otimes \mathbf{r},\, \mathbf{r} \in \mathbb{R}^2, \mathbf{G} \in \mathbb{R}^{2\times 2}\} \tag{1}$$

$$= \{\mathbf{q} \in \mathbb{R}^{2n} : \mathbf{q}_i = \mathbf{G}\mathbf{c}_i + \mathbf{r}, \mathbf{r} \in \mathbb{R}^2, \mathbf{G} \in \mathbb{R}^{2\times 2}, \forall i \in \mathcal{N}\} \tag{2}$$

$$= \{\mathbf{q} \in \mathbb{R}^{2n} : \mathbf{q}_i - \mathbf{q}_j = \mathbf{G}(\mathbf{c}_i - \mathbf{c}_j), \mathbf{G} \in \mathbb{R}^{2\times 2}, \forall i, j \in \mathcal{N}\}. \tag{3}$$

This is the set of all configurations that can be expressed as an arbitrary translation $\mathbf{r}$ and an arbitrary linear transformation $\mathbf{G}$ applied on the nominal configuration $\mathbf{c}$; i.e., the set of all the affine transformations of $\mathbf{c}$ [6]. Note that, as in [6], [12]–[21], we do not require $\mathbf{G}$ to be invertible.

Formation graph structure. We model robot interactions via a constant simple undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. Every node is associated with a robot, i.e., $\mathcal{V} = \mathcal{N}$. We use node and robot as equivalent terms for all graph-related matters. An edge $\{i, j\}$ existing in $\mathcal{E}$ means robots i and j know their relative positions in $\mathbf{c}$ and, at run time, each can measure the

other's current relative position. We denote by $\mathcal{N}_i \subset \mathcal{N}$ the set of neighbors of i in $\mathcal{G}$. Next, we give our definitions of two concepts used to specify the structure of $\mathcal{G}$.

Definition 2 (Module): A module $\mathcal{T}_l = \{l_i\}_{i=1}^4$, where $l_i \in \mathcal{N}\ \forall i$, is a set of four different robots such that there is a graph edge in $\mathcal{E}$ between every pair of robots in the set.

Note that, in graph terms, a module is a four-node clique. We will consider that $\mathcal{G}$ is organized in modules. We assume there are w distinct modules in $\mathcal{G}$ and, for simplicity, there are no edges outside modules: i.e., if two robots are graph neighbors, they are in a common module. Considering an arbitrary indexing of modules, we denote the set of modules by $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_w\}$. We also denote, for a robot i, the set of modules that i belongs to by $\mathcal{W}_i = \{\mathcal{T}_l \in \mathcal{T} : i \in \mathcal{T}_l\}$.

Definition 3 (Interlacing): We say any two different modules $\mathcal{T}_l$, $\mathcal{T}_m$ are *interlaced* if they have exactly three robots in common. Furthermore, we define a *path of interlacings* between any two different modules $\mathcal{T}_l$ and $\mathcal{T}_m$ as an ordered sequence of modules $\mathcal{P}_{lm} = (\mathcal{T}_l, \ldots, \mathcal{T}_m)$ such that every module is interlaced with the next one in the sequence.

Assumptions. Using the above definitions, we make two assumptions about the graph and the nominal configuration.

Assumption 1 (Structure of $\mathcal{G}$): $\mathcal{G}$ is such that every robot $i \in \mathcal{N}$ is in at least one module, and there exists a path of interlacings between every pair of modules in $\mathcal{T}$.

Assumption 2 (Positions in nominal configuration): There are no two equal positions in $\mathbf{c}$. Moreover, if three robots i, j, k are neighbors in $\mathcal{G}$, i.e., if $\{i, j\} \in \mathcal{E}$, $\{j, k\} \in \mathcal{E}$, $\{i, k\} \in \mathcal{E}$, then the nominal configuration positions $\mathbf{c}_i$, $\mathbf{c}_j$ and $\mathbf{c}_k$ do not form a straight line.

Let us emphasize that Assumption 2 refers to the constant nominal configuration ($\mathbf{c}$), not to the current configuration ($\mathbf{q}$). These two assumptions will guarantee that convergence of every module to a partial four-robot affine formation implies convergence of the full team to the n-robot affine formation. Our control strategy relies on this guarantee, as we will show later. The assumptions and an intuition behind our strategy are visualized with an example in Fig. 1. A three-robot interlacing makes two modules share the same affine transformation. This is why we need at least four robots per module. While modules with more robots could be used, we use minimum-size (i.e., four-robot) modules since this requires fewer graph edges and allows for a simpler and more uniform system design.

Prior works, e.g., [6], [12] require universal rigidity of the graph/framework. While this requirement can be achieved with sparser (i.e., having fewer edges) graphs than those satisfying Assumption 1, our graph structure still allows a scalable distributed architecture, and has the advantage of its modularity. For example, in Fig. 1, $w = n - 3$ and $|\mathcal{E}| = 3(w+1) = 3(n-2)$. Hence, the number of edges grows linearly with n, and is at most three per robot, which shows that the structure is scalable. Note that this same growth pattern (linear) occurs for rigid graphs, e.g., for Laman graphs [35]. Note, too, that prior works also considered assumptions about the nominal configuration (e.g., it being generic [6], [12]), as we do in Assumption 2. Being generic is a global property that depends on the full nominal configuration, whereas our Assumption 2 only defines positional restrictions for robots that are graph neighbors.

Problem statement. Our goal is to design a control law for $\mathbf{u}_i\ \forall i \in \mathcal{N}$ such that, under the described conditions, the multirobot team converges to an affine formation.

III. Modular affine formation control strategy

The core idea of the strategy we propose is to achieve an affine formation for every module. Thanks to Assumptions 1 and 2, this will imply achieving an affine formation for the full team, as will be shown in our analysis. Hence, we start by presenting the control design for a module.

A. Control strategy for a module

Consider a given module $\mathcal{T}_l = \{l_1, l_2, l_3, l_4\}$. We choose an arbitrary ordering of the robots in the module: e.g., (l_1, l_2, l_3, l_4), and group current and nominal robot positions in matrices of size 2×4, as

$$\mathbf{Q}_l = [\mathbf{q}_{l_1}, \mathbf{q}_{l_2}, \mathbf{q}_{l_3}, \mathbf{q}_{l_4}],\ \mathbf{C}_l = [\mathbf{c}_{l_1}, \mathbf{c}_{l_2}, \mathbf{c}_{l_3}, \mathbf{c}_{l_4}]. \quad (4)$$

Our goal is to define a controller to make the positions $\mathbf{Q}_l$ equal to an affine transformation of the positions $\mathbf{C}_l$. For this, we exploit least-squares affine shape alignment, as described next. Considering an affine transformation defined as a linear transformation plus a translation (see Definition 1), we seek to find $\mathbf{F}_l \in \mathbb{R}^{2\times 2}$ and $\mathbf{t}_l \in \mathbb{R}^2$ for which

$$\|\mathbf{Q}_l - \mathbf{F}_l\mathbf{C}_l - \mathbf{1}_4^\intercal \otimes \mathbf{t}_l\| \quad (5)$$

is minimum. For the translation part, it can be seen that this expression is minimized with $\mathbf{t}_l = \mathbf{0}$ when the centroids of $\mathbf{Q}_l$ and $\mathbf{C}_l$ coincide. Hence, for convenience, we will remove $\mathbf{t}_l$ and make the centroids of the two sets equal to zero. Specifically, let us define the centroids of $\mathbf{Q}_l$ and $\mathbf{C}_l$ as $\mathbf{g}_l^q = (1/4)\mathbf{Q}_l\mathbf{1}_4$ and $\mathbf{g}_l^c = (1/4)\mathbf{C}_l\mathbf{1}_4$, respectively. We then define centered versions of $\mathbf{Q}_l$ and $\mathbf{C}_l$ as

$$\begin{aligned}\mathbf{Q}_{lb} &= [\mathbf{q}_{l_1} - \mathbf{g}_l^q, \mathbf{q}_{l_2} - \mathbf{g}_l^q, \mathbf{q}_{l_3} - \mathbf{g}_l^q, \mathbf{q}_{l_4} - \mathbf{g}_l^q] = \mathbf{Q}_l\mathbf{K}_4,\\ \mathbf{C}_{lb} &= [\mathbf{c}_{l_1} - \mathbf{g}_l^c, \mathbf{c}_{l_2} - \mathbf{g}_l^c, \mathbf{c}_{l_3} - \mathbf{g}_l^c, \mathbf{c}_{l_4} - \mathbf{g}_l^c] = \mathbf{C}_l\mathbf{K}_4. \quad (6)\end{aligned}$$

We can now reformulate the optimization problem as finding

$$\mathbf{G}_l = \arg\min_{\mathbf{F}_l} \|\mathbf{Q}_{lb} - \mathbf{F}_l\mathbf{C}_{lb}\|. \quad (7)$$

As is well known, the problem (7) can be solved using the Moore-Penrose inverse $\mathbf{C}_{lb}^+$ by choosing

$$\mathbf{G}_l = \mathbf{Q}_{lb}\mathbf{C}_{lb}^+. \quad (8)$$

The robot positions in $\mathbf{C}_{lb}$ do not form a straight line, by Assumption 2. Therefore, the two rows of $\mathbf{C}_{lb}$ are linearly independent. Hence, the 2×2 matrix $\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal$ has rank two, i.e., it is invertible, and we can express

$$\mathbf{C}_{lb}^+ = \mathbf{C}_{lb}^\intercal(\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal)^{-1}. \quad (9)$$

Note that $\mathbf{C}_{lb}^+$ is constant. Let us now use $\mathbf{G}_l$ to define the following cost for the module:

$$\gamma_l = (1/2)\,\|\mathbf{Q}_{lb} - \mathbf{G}_l\mathbf{C}_{lb}\|^2. \quad (10)$$

Substituting $\mathbf{Q}_{lb} = \mathbf{Q}_l\mathbf{K}_4$, (8), and (9), we can write

$$\gamma_l = (1/2)\,\|\mathbf{Q}_l\mathbf{K}_4 - \mathbf{Q}_l\mathbf{K}_4\mathbf{C}_{lb}^\intercal(\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal)^{-1}\mathbf{C}_{lb}\|^2. \quad (11)$$

Notice that $\mathbf{K}_4$ is symmetric and idempotent, i.e., $\mathbf{K}_4^2 = \mathbf{K}_4\mathbf{K}_4 = \mathbf{K}_4$. Hence, the term $\mathbf{K}_4\mathbf{C}_{lb}^\intercal$ in (11) is

$$\mathbf{K}_4\mathbf{C}_{lb}^\intercal = (\mathbf{C}_{lb}\mathbf{K}_4)^\intercal = (\mathbf{C}_l\mathbf{K}_4\mathbf{K}_4)^\intercal = (\mathbf{C}_l\mathbf{K}_4)^\intercal = \mathbf{C}_{lb}^\intercal. \quad (12)$$

With these developments, we can now express γ_l as follows:

$$\gamma_l = (1/2)\,\|\mathbf{Q}_l\mathbf{E}_l\|^2, \quad (13)$$

where

$$\mathbf{E}_l = \mathbf{K}_4 - \mathbf{C}_{lb}^\intercal(\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal)^{-1}\mathbf{C}_{lb} \quad \in \mathbb{R}^{4\times 4}. \quad (14)$$

Observe that $\mathbf{E}_l$ is symmetric. Next, we will express the above in a vectorized formulation. For this, we define $\mathbf{q}_l = vec(\mathbf{Q}_l) = [\mathbf{q}_{l_1}^\intercal, \mathbf{q}_{l_2}^\intercal, \mathbf{q}_{l_3}^\intercal, \mathbf{q}_{l_4}^\intercal]^\intercal \in \mathbb{R}^8$ and

$$\mathbf{B}_l = \mathbf{E}_l \otimes \mathbf{I}_2 \quad \in \mathbb{R}^{8\times 8}. \quad (15)$$

Note that $\mathbf{B}_l$ is constant and symmetric. It is direct to see that, due to the symmetry of $\mathbf{E}_l$, the entries of the 8×1 vector $\mathbf{B}_l\mathbf{q}_l$ are equal to the entries of the 2×4 matrix $\mathbf{Q}_l\mathbf{E}_l$. Hence, we have $\|\mathbf{Q}_l\mathbf{E}_l\| = \|\mathbf{B}_l\mathbf{q}_l\|$. Notice as well that $\mathbf{E}_l^2 = \mathbf{E}_l$ and $\mathbf{B}_l^2 = \mathbf{B}_l$. Therefore, $\|\mathbf{Q}_l\mathbf{E}_l\|^2 = \|\mathbf{B}_l\mathbf{q}_l\|^2 = \mathbf{q}_l^\intercal\mathbf{B}_l^\intercal\mathbf{B}_l\mathbf{q}_l = \mathbf{q}_l^\intercal\mathbf{B}_l\mathbf{q}_l$. Hence, from (13), we reach

$$\gamma_l = (1/2)\mathbf{q}_l^\intercal\mathbf{B}_l\mathbf{q}_l. \quad (16)$$

Note that $\gamma_l \geq 0$ by definition (see, e.g., (13)). Therefore, $\mathbf{B}_l$ is symmetric positive semidefinite.

Next, we express γ_l in terms of the full stack vector $\mathbf{q}$, using a selection matrix. Recalling that the ordering we considered in (4) for the four robots in $\mathcal{T}_l$ is (l_1, l_2, l_3, l_4), we define $\bar{\mathbf{P}}_l \in \mathbb{R}^{4\times n}$ as a matrix with ones in the positions $\bar{\mathbf{P}}_l[1, l_1]$, $\bar{\mathbf{P}}_l[2, l_2]$, $\bar{\mathbf{P}}_l[3, l_3]$, $\bar{\mathbf{P}}_l[4, l_4]$ and zeros anywhere else. Then, we define our selection matrix as $\mathbf{P}_l = \bar{\mathbf{P}}_l \otimes \mathbf{I}_2 \in \mathbb{R}^{8\times 2n}$. Notice then that $\mathbf{q}_l = \mathbf{P}_l\mathbf{q}$ and that we can write

$$\mathbf{q}_l^\intercal\mathbf{B}_l\mathbf{q}_l = \mathbf{q}^\intercal\mathbf{P}_l^\intercal\mathbf{B}_l\mathbf{P}_l\mathbf{q} = \mathbf{q}^\intercal\mathbf{A}_l\mathbf{q}, \quad (17)$$

with $\mathbf{A}_l = \mathbf{P}_l^\intercal\mathbf{B}_l\mathbf{P}_l$. Observe that $\mathbf{A}_l \in \mathbb{R}^{2n\times 2n}$ is constant and symmetric. Using (17), γ_l in (16) can be expressed as

$$\gamma_l = (1/2)\mathbf{q}^\intercal\mathbf{A}_l\mathbf{q}. \quad (18)$$

Note that $\mathbf{A}_l$ is positive semidefinite. This expression of γ_l as a function of the stack-form state $\mathbf{q}$ will be exploited for defining and analyzing our proposed controller. From (10), if the robots in the module $\mathcal{T}_l$ move in a way that drives γ_l to zero, this makes $\mathbf{Q}_{lb} = \mathbf{G}_l\mathbf{C}_{lb}$; which implies, from (2), that $\mathcal{T}_l$ is in an affine formation. Therefore, we propose to control the full team to make $\gamma_l = 0$ for all l, as seen in Sec. III-B.

Remark 1: Note that, among all possible affine formations for the robots in $\mathcal{T}_l$, we make the robots move toward the one that minimizes (5) (i.e., is closest to the current robot positions $\mathbf{Q}_l$ in least-squares sense). This is interesting as it means the formation can be reached faster and with shorter traveled distances, i.e., more efficiently.

B. Proposed control law

We define an aggregate cost for the full team as

$$\gamma = \sum_{l=1}^{w}\gamma_l = (1/2)\mathbf{q}^\intercal\mathbf{A}\mathbf{q}, \quad (19)$$

where, from (18), the matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \sum_{l=1}^{w}\mathbf{A}_l \quad \in \mathbb{R}^{2n\times 2n}. \quad (20)$$

Notice that $\mathbf{A}$ is constant and symmetric. It is also positive semidefinite, as $\gamma \geq 0$. As said above, our idea is to drive γ_l to zero for all l. To this end, our strategy is to move $\mathbf{q}$ along the negative gradient of γ. Notice that this equals the sum of negative gradients for all γ_l. Therefore, from the perspective of a given robot i, this strategy is equivalent to computing a negative gradient vector for every module $\mathcal{T}_l$ that i is a part of, and adding these vectors up. This is explained in more detail in Sec. III-C. The negative gradient of γ has the form

$$-\nabla_{\mathbf{q}}\gamma = -\mathbf{A}\mathbf{q}. \quad (21)$$

The proposed control law, where $\kappa > 0$ is a scalar gain, is

$$\mathbf{u} = -\kappa\mathbf{A}\mathbf{q}. \quad (22)$$

Remark 2: We will use the full-team control expression (22) for our analysis (Sec. IV). However, let us emphasize that knowledge of matrix $\mathbf{A}$ is not needed for using our approach. Our developments above show how to obtain the specific form of this matrix, which ultimately depends on the nominal configuration and the module-based graph structure. One can see that $\mathbf{A}$ can be expressed as $\mathbf{A} = \bar{\mathbf{A}} \otimes \mathbf{I}_2$, it is symmetric positive semidefinite, and its kernel is the affine image of the nominal configuration (as will be shown in Proposition 1). Hence, $\bar{\mathbf{A}}$ is the equivalent of the stress matrix used in the control design of other works, e.g., [6], [13]–[16], [20], [21]. The key difference is that, by exploiting modularity, our approach does not use at any stage of its design or implementation the values or properties of $\mathbf{A}$. The next section demonstrates this by explaining how each robot can implement our approach.

C. Implementation of the controller

We describe a possible implementation procedure for a given robot i in Algorithm 1. In the algorithm, a superindex i is used for variables used internally by robot i, i.e., variables whose values do not need to be known by other robots or agreed upon with other robots. $\mathbf{q}_j^i$ in the algorithm denotes the measured position of j in the reference frame of i. It can be shown that the reference frame of i can be completely independent from the reference frames of the other robots. This property is common to other affine formation control approaches (see, e.g., [6, Sec. V-A.1]) and it provides the important advantage of facilitating the use of onboard sensors, e.g., cameras, by the robots.

In the *Control execution phase*, robot i uses only the identities and relative positions of the robots in the modules that i belongs to. Therefore, the approach is distributed and

global information is not needed. Another important fact is that the robots do not need to share any measurements: i.e., communications are not required in this phase. A fundamental advantage of the proposed modular approach is that the *Design/reconfiguration phase* can be performed locally. Indeed, it is possible to design the formation control system module-by-module, by simply defining the needed connections (i.e., interlacings) between modules. The same strategy can be used for reconfiguration: i.e., if the task requires local changes in the graph, the nominal configuration, or the number of robots, this can be handled locally by the involved robots, unlike with approaches that rely on global system properties or centralized designs. Communication-based initialization/reconfiguration procedures have to be used allowing robots to coordinately switch between phases. Needing such procedures is common in formation control and their specifics exceed the scope of the problem addressed in this article.

Algorithm 1 Implementation of the controller by robot $i \in \mathcal{N}$

Design/reconfiguration phase:
Determine $\mathcal{N}_i$ and $\mathcal{W}_i$
for $\mathcal{T}_l \in \mathcal{W}_i$ **do**
  Choosing arbitrarily an order for the other three robots in $\mathcal{T}_l$, define the ordering $(l_1^i, l_2^i, l_3^i, l_4^i)$ with $l_1^i = i$
  Determine, from relative positions $\mathbf{c}_{l_j^i} - \mathbf{c}_{l_k^i}$ for j, k in $\{1, \ldots, 4\}$, $\mathbf{C}_{lb}^i$ (6), compute $\mathbf{B}_l^i$ (15), and store the first two rows of $\mathbf{B}_l^i$ as matrix $\mathbf{B}_{l2}^i$ of size 2×8
end for
Control execution phase:
while *Control task is running* **do**
  Define $\mathbf{d}_i^i = [0, 0]^\intercal$
  Measure $\mathbf{q}_j^i$, $\forall j \in (\mathcal{N}_i \cup i)$
  for $\mathcal{T}_l \in \mathcal{W}_i$ **do**
    Form $\mathbf{q}_l^i = [\mathbf{q}_i^{i\intercal}, \mathbf{q}_{l_2^i}^{i\intercal}, \mathbf{q}_{l_3^i}^{i\intercal}, \mathbf{q}_{l_4^i}^{i\intercal}]^\intercal$
    Compute $\mathbf{d}_{il}^i = -\mathbf{B}_{l2}^i \mathbf{q}_l^i$
    Update $\mathbf{d}_i^i = \mathbf{d}_i^i + \mathbf{d}_{il}^i$
  end for
  Apply control $\mathbf{u}_i^i = \kappa \mathbf{d}_i^i$
end while

IV. Stability analysis

We first establish, in Proposition 1, the link between affine formation achievement and the kernel of matrix $\mathbf{A}$.

Proposition 1 (Affine formation condition from $\mathbf{A}$*):* Under Assumptions 1 and 2, the multirobot team is in an affine formation if and only if $\mathbf{Aq} = \mathbf{0}$.

Proof: (Necessity) Suppose the team is in an affine formation. Hence, from (1), there exist a $\mathbf{G} \in \mathbb{R}^{2\times 2}$ and an $\mathbf{r} \in \mathbb{R}^2$ such that $\mathbf{q} = (\mathbf{I}_n \otimes \mathbf{G})\mathbf{c} + \mathbf{1}_n \otimes \mathbf{r}$. Consider a given module $\mathcal{T}_l$. Expressing the robot positions in $\mathcal{T}_l$ in 2×4 format, we have $\mathbf{Q}_l = \mathbf{GC}_l + \mathbf{1}_4^\intercal \otimes \mathbf{r}$. Post-multiplying by $\mathbf{K}_4$ gives $\mathbf{Q}_{lb} = \mathbf{GC}_{lb} + (\mathbf{1}_4^\intercal \otimes \mathbf{r})\mathbf{K}_4$ where, as $\mathbf{1}_4^\intercal \otimes \mathbf{r} = [\mathbf{r}, \mathbf{r}, \mathbf{r}, \mathbf{r}]$ and $\mathbf{K}_4 = \mathbf{I}_4 - (1/4)\mathbf{1}_4\mathbf{1}_4^\intercal$, $(\mathbf{1}_4^\intercal \otimes \mathbf{r})\mathbf{K}_4$ is equal to $\mathbf{0}$.

Hence, we have $\mathbf{Q}_{lb} = \mathbf{GC}_{lb}$. Substituting this expression of $\mathbf{Q}_{lb}$ in the definition of $\mathbf{G}_l$ (8) and using (9), we obtain

$$\mathbf{G}_l = \mathbf{Q}_{lb}\mathbf{C}_{lb}^\intercal(\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal)^{-1} = \mathbf{GC}_{lb}\mathbf{C}_{lb}^\intercal(\mathbf{C}_{lb}\mathbf{C}_{lb}^\intercal)^{-1} = \mathbf{G},$$

which holds for all modules l. Therefore, as $\mathbf{G} = \mathbf{G}_l$, it holds that $\mathbf{Q}_{lb} = \mathbf{G}_l\mathbf{C}_{lb}$ $\forall l$, and consequently, from (10), $\gamma_l = 0$ $\forall l$. Hence, from (19), $\gamma = 0$, i.e., $\mathbf{q}^\intercal\mathbf{Aq} = 0$. As $\mathbf{A}$ is real symmetric positive semidefinite, there exists a real matrix $\mathbf{F}$ such that $\mathbf{A} = \mathbf{F}^\intercal\mathbf{F}$. Therefore $\mathbf{q}^\intercal\mathbf{Aq} = \mathbf{q}^\intercal\mathbf{F}^\intercal\mathbf{Fq} = \|\mathbf{Fq}\|^2 = 0$, which implies $\mathbf{Fq} = \mathbf{0}$ and hence $\mathbf{Aq} = \mathbf{0}$.

(Sufficiency) Now, suppose $\mathbf{Aq} = \mathbf{0}$. This implies $\gamma = 0$ from (19), which in turn implies $\gamma_l = 0$ $\forall l$. Therefore, due to (10), we have $\mathbf{Q}_{lb} = \mathbf{G}_l\mathbf{C}_{lb}$ $\forall l$, where $\mathbf{G}_l$ is the transformation corresponding to module l. Let us take two of the robots in $\mathcal{T}_l$: for example, l_1 and l_2. Notice from (6) that we have $\mathbf{q}_{l_1} - \mathbf{g}_l^q = \mathbf{G}_l(\mathbf{c}_{l_1} - \mathbf{g}_l^c)$ and $\mathbf{q}_{l_2} - \mathbf{g}_l^q = \mathbf{G}_l(\mathbf{c}_{l_2} - \mathbf{g}_l^c)$. Subtracting these two equations, we get the expression

$$\mathbf{q}_{l_1} - \mathbf{q}_{l_2} = \mathbf{G}_l(\mathbf{c}_{l_1} - \mathbf{c}_{l_2}). \tag{23}$$

Note that an analogous expression holds for every pair of robots in every module. Consider two interlaced modules r and s, and call the three robots they share i, j and k. As $\mathbf{Q}_{rb} = \mathbf{G}_r\mathbf{C}_{rb}$ and $\mathbf{Q}_{sb} = \mathbf{G}_s\mathbf{C}_{sb}$, applying (23) we get

$$\begin{aligned} \mathbf{q}_i - \mathbf{q}_j &= \mathbf{G}_r(\mathbf{c}_i - \mathbf{c}_j) = \mathbf{G}_s(\mathbf{c}_i - \mathbf{c}_j), \\ \mathbf{q}_i - \mathbf{q}_k &= \mathbf{G}_r(\mathbf{c}_i - \mathbf{c}_k) = \mathbf{G}_s(\mathbf{c}_i - \mathbf{c}_k). \end{aligned} \tag{24}$$

Let us define $\mathbf{c}_{ij} = \mathbf{c}_i - \mathbf{c}_j$, $\mathbf{c}_{ik} = \mathbf{c}_i - \mathbf{c}_k$, $\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j$ and $\mathbf{q}_{ik} = \mathbf{q}_i - \mathbf{q}_k$. Notice that we have

$$[\mathbf{q}_{ij}, \mathbf{q}_{ik}] = \mathbf{G}_r[\mathbf{c}_{ij}, \mathbf{c}_{ik}] = \mathbf{G}_s[\mathbf{c}_{ij}, \mathbf{c}_{ik}]. \tag{25}$$

Recall that there are no two equal positions in $\mathbf{c}$ per Assumption 2, i.e., we have $\mathbf{c}_{ij} \neq \mathbf{0}$, $\mathbf{c}_{ik} \neq \mathbf{0}$. Observe that $\det([\mathbf{c}_{ij}, \mathbf{c}_{ik}])$ can only be zero if $\mathbf{c}_{ij} = \beta\mathbf{c}_{ik}$ for some $\beta \in \mathbb{R}$, i.e., if the positions i, j and k in the nominal configuration $\mathbf{c}$ form a straight line. This is not possible, due to Assumption 2. Therefore, $[\mathbf{c}_{ij}, \mathbf{c}_{ik}]^{-1}$ exists and hence $\mathbf{G}_r$, $\mathbf{G}_s$ are unique and identical, satisfying $\mathbf{G}_r = \mathbf{G}_s = [\mathbf{q}_{ij}, \mathbf{q}_{ik}][\mathbf{c}_{ij}, \mathbf{c}_{ik}]^{-1}$.

Therefore, $\mathbf{G}_r = \mathbf{G}_s$ for any two interlaced modules r and s. By induction, as any two modules can be joined by successive interlacings due to Assumption 1, we conclude that there is a $\mathbf{G}$ such that $\mathbf{G}_l = \mathbf{G}$ $\forall l$. Therefore, $\mathbf{Q}_{lb} = \mathbf{GC}_{lb}$ $\forall l$. As shown earlier in the proof via (23), this implies, for every module, that $\mathbf{q}_i - \mathbf{q}_j = \mathbf{G}(\mathbf{c}_i - \mathbf{c}_j)$ for every pair i, j in the module. Then, Assumption 1 also implies that module interlacings create a path in $\mathcal{G}$ between every pair of nodes i, $j \in \mathcal{N}$ which ensures, again applying induction, that $\mathbf{q}_i - \mathbf{q}_j = \mathbf{G}(\mathbf{c}_i - \mathbf{c}_j)$ holds for every pair i, $j \in \mathcal{N}$. Therefore, from (3), the team is in an affine formation. ■

As seen in the proof, $\gamma = 0 \Leftrightarrow \mathbf{Aq} = \mathbf{0}$ as $\mathbf{A}$ is symmetric positive semidefinite. This gives the following corollary.

Corollary 1: Under Assumptions 1 and 2, the multirobot team is an affine formation if and only if $\gamma = 0$.

Remark 3: The reason why we assume three-robot interlacings is clear from the proof of Proposition 1, and visualized in Fig. 1: under Assumption 2, if a module forms an affine transformation of the nominal configuration, three robot positions determine uniquely the transformation. Hence, for any two modules, if 1) each forms an affine transformation individually and 2) the two have three robots in common, then the affine transformation is the same for both. This is not true if there are only two common robots.

Next, we show the convergence of our control law.

Theorem 1 (Convergence to affine formation): Using the control (22) the multirobot team converges asymptotically to a static affine formation. Moreover, the centroid is preserved, and $\mathbf{q}(t \to \infty)$ is equal to the orthogonal projection of the initial configuration $\mathbf{q}(t=0)$ onto the kernel of $\mathbf{A}$.

Proof: Under (22), the dynamics is $\dot{\mathbf{q}} = -\kappa \mathbf{A}\mathbf{q}$. As $\mathbf{A}$ is symmetric positive semidefinite, from well-known results (see, e.g., [14, Lemma 1]) $\mathbf{q}$ converges asymptotically to the kernel of $\mathbf{A}$; in particular, to a static configuration equal to the orthogonal projection of the initial configuration $\mathbf{q}(t=0)$ onto the kernel of $\mathbf{A}$. Therefore $\mathbf{A}\mathbf{q} \to \mathbf{0}$ as $t \to \infty$. From Proposition 1, the team converges to an affine formation.

The centroid of the team can be defined as $\mathbf{g}_\mathbf{q} = (1/n)(\mathbf{1}_n \otimes \mathbf{I}_2)^\intercal \mathbf{q}$. We can then obtain the dynamics of the centroid under the action of the control (22) as

$$\dot{\mathbf{g}}_\mathbf{q} = (1/n)(\mathbf{1}_n \otimes \mathbf{I}_2)^\intercal \dot{\mathbf{q}} = (-\kappa/n)(\mathbf{1}_n \otimes \mathbf{I}_2)^\intercal \mathbf{A}\mathbf{q}. \quad (26)$$

Following [14, Lemma 3], we notice that $((\mathbf{1}_n \otimes \mathbf{I}_2)^\intercal \mathbf{A})^\intercal = \mathbf{A}(\mathbf{1}_n \otimes \mathbf{I}_2) = [\mathbf{A}(\mathbf{1}_n \otimes [1,0]^\intercal), \mathbf{A}(\mathbf{1}_n \otimes [0,1]^\intercal)]$. Clearly, $\mathbf{q} = \mathbf{1}_n \otimes [1,0]^\intercal$ and $\mathbf{q} = \mathbf{1}_n \otimes [0,1]^\intercal$ are affine formations, as they satisfy (1) by choosing $\mathbf{G} = \mathbf{0}$ and, respectively, $\mathbf{r} = [1,0]^\intercal$ and $\mathbf{r} = [0,1]^\intercal$. Therefore, from Proposition 1, $[\mathbf{A}(\mathbf{1}_n \otimes [1,0]^\intercal), \mathbf{A}(\mathbf{1}_n \otimes [0,1]^\intercal)] = \mathbf{0}$ and we can conclude from (26) that $\dot{\mathbf{g}}_\mathbf{q} = \mathbf{0}$, i.e., the centroid is preserved. ■

Remark 4: Centroid preservation and convergence to the orthogonal projection of the initial configuration are interesting properties for motion efficiency. Having affine formations where all robots are in the same position ($\mathbf{G} = \mathbf{0}$ in Definition 1) is theoretically possible in degenerate cases. In practice, this possibility is excluded, e.g., by control design via leader robots or leaderless maneuvers [6], [12], [13], [16], [20], [21] or by inherent constraints of a higher-level mission, e.g., target enclosing. As the law (22) does not control specifically the values in $\mathbf{G}$ and $\mathbf{r}$ of Definition 1, these parameters are adaptable during a mission.

V. Multitarget enclosing and tracking exploiting modular affine formation control

In this section, we propose to apply our modular affine formation controller to the task of multitarget enclosing and tracking. This task is important for scenarios such as tracking of human subjects [33], [34] and has been addressed with formation-based approaches [4], [5], [31], [32]. We will introduce an additional control action to enclose a set of moving targets, while trying to keep an affine formation via the control approach presented above. This approach is a good choice for an enclosing task due to the adaptability of the parameters in $\mathbf{G}$ and $\mathbf{r}$, and the higher flexibility of affine formations compared to other formation types. For an enclosing task, natural nominal configurations include mainly circular (i.e., regular polygons) or other polygonal formations. We assume there are m point targets in the workspace and a target can be perceived by a robot if it is located within a detection radius ρ. At any instant, every robot i has a set of m_i targets within a ρ radius; we denote their positions by $\mathbf{x}_i = \{\mathbf{x}_{1i}, \ldots, \mathbf{x}_{mi}\}$ and their centroid by $\bar{\mathbf{x}}_i$. These sets of targets can be time-varying. We propose to achieve enclosing by exploiting the convex hull of $\mathbf{x}_i$, denoted by $\mathcal{X}_i$. We enlarge this convex hull by extending every vector from the centroid to every vertex a predefined distance, and denote the enlarged convex hull by $\mathcal{X}_i^e$. The enclosing action we define for each robot i has two possibilities:

1) If $\mathbf{q}_i$ is outside $\mathcal{X}_i^e$, the robot moves toward the closest point in $\mathcal{X}_i^e$.
2) If $\mathbf{q}_i$ is inside $\mathcal{X}_i^e$, the robot moves away from $\bar{\mathbf{x}}_i$ along the line that joins $\mathbf{q}_i$ to $\bar{\mathbf{x}}_i$, trying to leave $\mathcal{X}_i^e$.

The idea of this strategy is to fully enclose all targets by continuously moving the robots toward the enlarged convex hulls. The strategy is more reactive when the enclosing is lost, i.e., item 2) above. The strategy is distributed as each robot only considers those targets within its perceptual range. We enlarge the convex hull to give the robots more space to react to escaping motions by the targets, and to prevent robot-target collisions. We assume the velocities of the targets are not larger than the achievable robot velocities, which is reasonable in the context of a tracking mission.

The described strategy generates an enclosing control vector $\mathbf{d}_i^e$ for every robot i. To combine this with the affine formation control action, we use a weighted sum of the two. Concretely, the total control action, $\mathbf{u}_i^t$, for a robot i is

$$\mathbf{u}_i^t = \kappa_p(w_a \mathbf{d}_i + w_e \mathbf{d}_i^e), \quad \forall i \in \mathcal{N}, \quad (27)$$

where w_a, $w_e = 1 - w_a$ are non-negative scalar weights, κ_p is a positive scalar gain, and $\mathbf{d}_i$ is the control vector from Sec. III-B, defined in Algorithm 1. While we do not offer formal performance guarantees for the controller in (27), we verify its practical usefulness experimentally in the next section.

VI. Experimental validation

We start by presenting results from simulations, all run in MATLAB. First, we report on two tests of the modular affine formation controller, with results illustrated in Fig. 2. In both cases we implement Algorithm 1. Notice that in the two tests $\gamma \to 0$, i.e., the robots converge to an affine formation. In the first test, we use 10 robots. The nominal configuration consists of square-shaped modules forming an open chain as in Fig. 1, with $\mathbf{c}_1 = [0,0]^\intercal$, $\mathbf{c}_2 = [0,1]^\intercal$ in meters. At the final configuration shown in the x-y plot the matrix in (3) is $\mathbf{G} = [[0.381, -0.056]^\intercal, [0.281, 0.241]^\intercal]$ and the initial centroid is preserved. Hence, the final configuration is the result of translating, rotating, scaling, and shearing the nominal one. The second test is with 15 robots, a circular nominal configuration, and modules forming a closed chain. The centroid is preserved and the team achieves an ellipse.

Table I shows a comparative evaluation of traveled Euclidean distances until formation convergence (determined via an error threshold) for the 7-robot nominal configuration in [6, Fig. 3]. We test the stress-matrix-based leaderless approach mentioned after Eq. 12 of [6], and our approach with three graph structures (S1, S2, S3) where S1 is chained and S2, S3 are not. We compute mean and standard deviation over all robots using 10^4 runs from random initial positions uniformly distributed in a $5 \times 5\,\mathrm{m}^2$ area. The results show that the differences among the four approaches are not major.

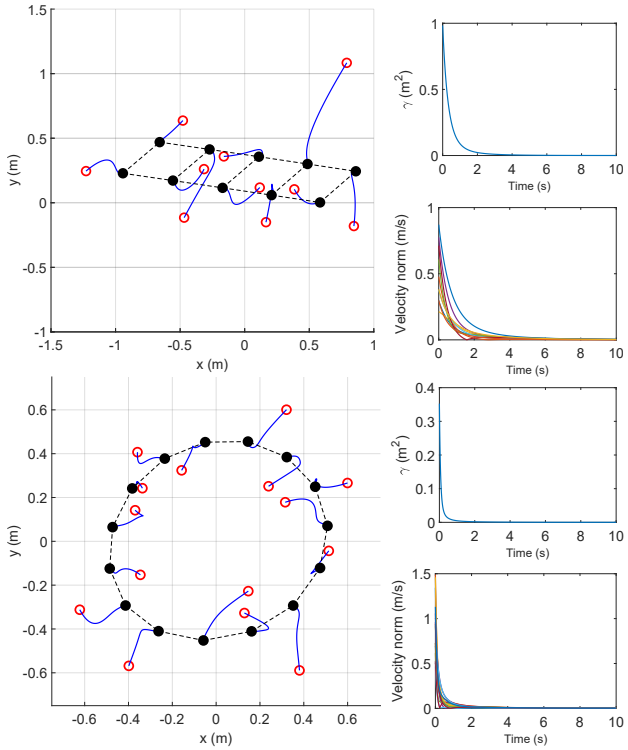

Fig. 2. Simulation results of affine formation control. Top part: 10-robot test. Bottom part: 15-robot test. In the x-y plots, robots move from initial (red hollow circles) to final (black full circles) positions following paths shown as blue lines. Final positions are at $t = 50\,\mathrm{s}$ (top) and $t = 20\,\mathrm{s}$ (bottom). The evolutions of γ and of the velocity norms are also shown.

TABLE I
TRAVELED DISTANCE TO REACH THE AFFINE FORMATION

Method	Mean (m)	St. dev. (m)	Modules
Stress [6]	1.509	0.171	–
S1	1.611	0.183	{1,2,3,4},{2,3,4,5},{3,4,5,6},{4,5,6,7}
S2	1.455	0.143	{1,6,7,2},{1,6,7,3},{1,6,7,4},{1,6,7,5}
S3	1.547	0.153	{1,4,5,3},{1,4,5,6},{1,4,5,7},{4,5,7,2}

Next, we test the multitarget enclosing and tracking approach of Sec. V. We use a circular nominal configuration and a closed chain of modules. We report on two tests, with $(w_a = 0.9, w_e = 0.1)$ and without $(w_a = 0, w_e = 1)$ the affine formation control action in (27). In these two tests we use saturation of the robot velocities. Some snapshots are shown in Fig. 3, and the complete execution is shown in the supplementary video. When using the affine control action, the robots can acquire and maintain a suitable enclosing configuration. Targets initially outside of the team's perimeter can get eventually enclosed. The team adapts to the varying shape formed by the set of targets and tracks persistent target displacements, with the robots being properly distributed. In contrast, when the affine control action is turned off, robots maintain a more irregular shape and leave some targets not enclosed, and multiple robots converge to the same position (observe the reduction in the number of black circles in the top row of Fig. 3). Note that we do not implement any collision avoidance mechanism. The values of γ indicate that with the affine control action, the team stays closer to an ellipse, which has the effect of deploying the robots more evenly. The approach in (27) should be considered a proof of concept,

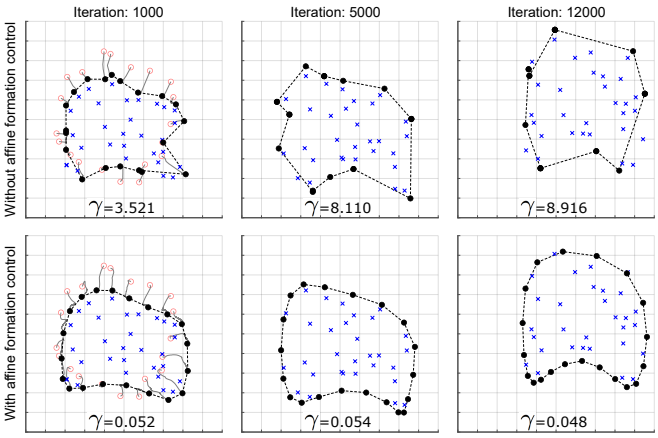

Fig. 3. Simulation results of multitarget enclosing and tracking, with 20 robots (black full circles) and 30 targets (blue crosses). We show three snapshots for two tests, without and with our proposed affine formation control, in the first and second row respectively. In both simulations, initial robot positions and target movements are the same. In the first column (iteration 1000), initial positions and robot paths are shown. γ in m^2 for each snapshot is also displayed. The cells in the plotted grid have size $1 \times 1\,\mathrm{m}^2$.

suitable for simplified scenarios; it may not provide a suitable enclosing in more challenging scenarios.

We finally discuss two tests done with physical robots in the Robotarium [36], illustrated in Fig. 4. As the robots have unicycle kinematics, we compute the single-integrator control (22) and use built-in conversion routines to generate the linear and angular unicycle velocity commands. In the first test we use 9 robots, the nominal configuration is a square inside a regular pentagon, and the modules form a closed chain. The robots converge to a static affine formation as illustrated by the final image and the curves of the velocities and γ. In the second test, we implement multitarget enclosing and tracking with 10 robots, a circular nominal configuration and a closed chain of modules. The robots manage to keep the moving targets enclosed, with the affine control action maintaining the team's shape close to an ellipse throughout the execution. These real-world experiments show that our control strategies, designed for single integrators, perform robustly when implemented on different dynamics (unicycles), and are applicable under non-ideal conditions, e.g., imperfect measurement and actuation.

VII. CONCLUSION

The modular multirobot controller proposed in this article allows achieving affine formations with a simple design process. Among the possible extensions and improvements of the presented approach, we are interested in 1) controlling the convergence time, 2) considering collision avoidance, and 3) taking further steps in the integration of affine formation control within multitarget perception tasks.

REFERENCES

[1] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *Int. J. Control*, vol. 82, no. 3, pp. 423–439, 2009.

[2] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex Laplacian," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1765–1777, 2014.

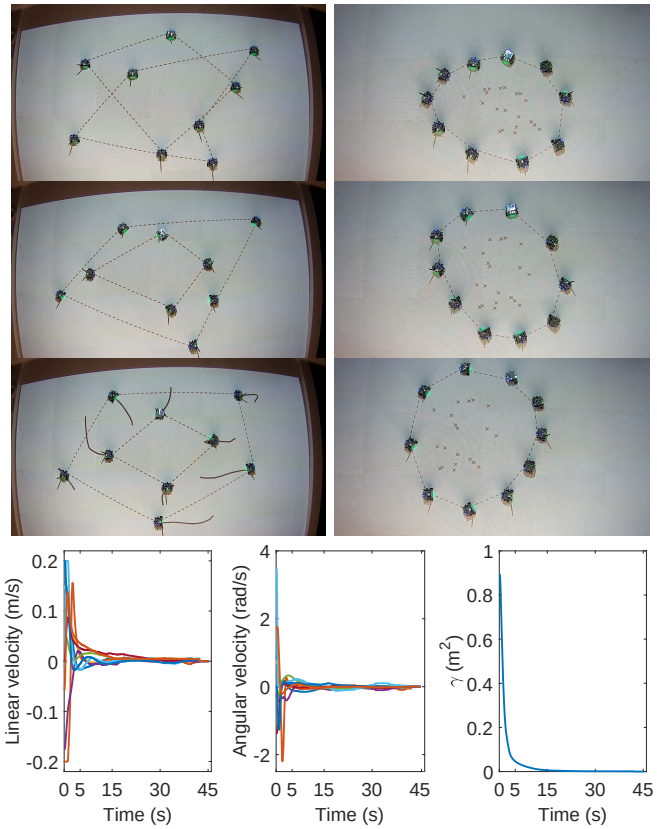

Fig. 4. Robotarium results. Rows 1 to 3 on the left: initial, intermediary, and final snapshots during modular affine formation control with 9 robots. Full robot paths are shown in row 3. Rows 1 to 3 on the right: three successive snapshots during multitarget enclosing with 10 robots and 20 targets (marked as crosses). Bottom row: evolution of robot velocities and γ in the 9-robot test. The full execution of these tests is shown in the supplementary video.

[3] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[4] L. Briñón-Arranz, A. Seuret, and A. Pascoal, "Circular formation control for cooperative target tracking with limited information," *J. Frankl. Inst.*, vol. 356, no. 4, pp. 1771–1788, 2019.

[5] G. López-Nicolás, M. Aranda, and Y. Mezouar, "Adaptive multirobot formation planning to enclose and track a target with motion and visibility constraints," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 142–156, 2020.

[6] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Trans. Automat. Control*, vol. 63, no. 12, pp. 4140–4155, 2018.

[7] Y. Liu *et al.*, "A distributed control approach to formation balancing and maneuvering of multiple multirotor UAVs," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 870–882, 2018.

[8] J. Fu, G. Wen, X. Yu, and Z.-G. Wu, "Distributed formation navigation of constrained second-order multiagent systems with collision avoidance and connectivity maintenance," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2149–2162, 2022.

[9] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *Int. J. Rob. Res.*, vol. 36, no. 9, pp. 1000–1021, 2017.

[10] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6466–6472, 2021.

[11] R. Herguedas, M. Aranda, G. López-Nicolás, C. Sagüés, and Y. Mezouar, "Double-integrator multirobot control with uncoupled dynamics for transport of deformable objects," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7623–7630, 2023.

[12] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, "Necessary and sufficient graphical conditions for affine formation control," *IEEE Trans. Automat. Control*, vol. 61, no. 10, pp. 2877–2891, 2016.

[13] H. Garcia de Marina, J. Jimenez Castellanos, and W. Yao, "Leaderless collective motions in affine formation control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6433–6438, 2021.

[14] M. Aranda, R. Aragüés, and G. López-Nicolás, "Combined leaderless control of translational, shape-preserving, and affine multirobot formations," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7567–7574, 2023.

[15] Y. Lin, Z. Lin, Z. Sun, and B. D. O. Anderson, "A unified approach for finite-time global stabilization of affine, rigid, and translational formation," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 1869–1881, 2022.

[16] L. Chen, J. Mei, C. Li, and G. Ma, "Distributed leader–follower affine formation maneuver control for high-order multiagent systems," *IEEE Trans. Automat. Control*, vol. 65, no. 11, pp. 4941–4948, 2020.

[17] Y. Xu, S. Zhao, D. Luo, and Y. You, "Affine formation maneuver control of high-order multi-agent systems over directed networks," *Automatica*, vol. 118, p. 109004, 2020.

[18] D. Li, G. Ma, Y. Xu, W. He, and S. S. Ge, "Layered affine formation control of networked uncertain systems: A fully distributed approach over directed graphs," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6119–6130, 2021.

[19] Y. Xu, D. Li, D. Luo, Y. You, and H. Duan, "Two-layer distributed hybrid affine formation control of networked Euler–Lagrange systems," *J. Frankl. Inst.*, vol. 356, no. 4, pp. 2172–2197, 2019.

[20] Q. Yang, H. Fang, M. Cao, and J. Chen, "Planar affine formation stabilization via parameter estimations," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5322–5332, 2022.

[21] K. Gao, Y. Liu, Y. Zhou, Y. Zhao, and P. Huang, "Practical fixed-time affine formation for multi-agent systems with time-based generators," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 69, no. 11, pp. 4433–4437, 2022.

[22] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory Appl.*, vol. 1, no. 2, pp. 505–512, 2007.

[23] R. Aldana-López, D. Gómez-Gutiérrez, R. Aragüés, and C. Sagüés, "Dynamic consensus with prescribed convergence time for multileader formation tracking," *IEEE Control Syst. Lett.*, vol. 6, pp. 3014–3019, 2022.

[24] H. García de Marina, M. Cao, and B. Jayawardhana, "Controlling rigid formations of mobile agents under inconsistent measurements," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 31–39, 2015.

[25] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos, "Distributed formation stabilization using relative position measurements in local coordinates," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3925–3935, 2016.

[26] K. Fathian, S. Safaoui, T. H. Summers, and N. R. Gans, "Robust distributed planar formation control for higher order holonomic and nonholonomic agents," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 185–205, 2021.

[27] M. Aranda, G. López-Nicolás, and Y. Mezouar, "Distributed linear control of multirobot formations organized in triads," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8498–8505, 2021.

[28] K. Sakurama, S.-I. Azuma, and T. Sugie, "Multiagent coordination via distributed pattern matching," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3210–3225, 2019.

[29] X. Yu, D. Saldaña, D. Shishika, and M. A. Hsieh, "Resilient consensus in robot swarms with periodic motion and intermittent communication," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 110–125, 2022.

[30] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, "Meshless deformations based on shape matching," *ACM Trans. Graph.*, vol. 24, no. 3, p. 471–478, 2005.

[31] M. Aranda, "Formation control on planar closed curves using Fourier descriptors," *IEEE Control Syst. Lett.*, vol. 7, pp. 3391–3396, 2023.

[32] T. Nägeli, S. Oberholzer, S. Plüss, J. Alonso-Mora, and O. Hilliges, "Flycon: Real-time environment-independent multi-view human pose estimation with aerial vehicles," *ACM Trans. Graph.*, vol. 37, no. 6, 2018.

[33] A. Pirinen, E. Gärtner, and C. Sminchisescu, "Domes to drones: Self-supervised active triangulation for 3D human pose reconstruction," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[34] A. Perez-Yus and A. Agudo, "Matching and recovering 3D people from multiple views," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1184–1193, 2022.

[35] X. Chen, M.-A. Belabbas, and T. Başar, "Global stabilization of triangulated formations," *SIAM J. Control. Optim.*, vol. 55, no. 1, pp. 172–199, 2017.

[36] S. Wilson *et al.*, "The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Syst. Mag.*, vol. 40, no. 1, pp. 26–44, 2020.