Contents

List of Figures

1

Flexible and steady control for cooperative target observation

Miguel Aranda and Youcef Mezouar

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France.
Email: {miguel.aranda,youcef.mezouar}@sigma-clermont.fr.

This work was supported by the French Government via FUI program (project Aerostrip) and Investissements d'avenir program (I-SITE project CAP 20-25 - MaRoC).

CONTENTS

This chapter considers the problem of coordinating multiple mobile robots to sense and track a moving target. For this task, it is equally fundamental to take into account the quality of the acquired target localization information and the suitability of the motions of the robotic team. A multirobot control methodology that simultaneously addresses these two concerns is described. Accurate and complete cooperative observation of the target is obtained by driving the robots to achieve a set of prescribed relative viewing angles, encap-

sulated by a default desired enclosing pattern, with respect to the target. In particular, relative robot position regulation and efficient target tracking are integrated via a formation-based controller that relies on global information and incorporates an optimal pattern rotation. The control framework allows each robot to select freely its desired distance to the target. By doing so, the robots can optimize their individual perception quality and avoid collisions during navigation without affecting the desired cooperative target observation diversity. It is shown that even with these distributed distance adjustments, the team movements remain steady with the proposed controller. This noteworthy property results in efficient agent robot motions and facilitates navigation and stable perception of the target by the team. An additional advantage of the described controller is that each robot can implement it using its independent local reference frame. Simulation tests in different cooperative target enclosing and navigation scenarios are presented to illustrate the performance of the methodology.

1.1 Introduction

It is well-known that a multirobot system can allow to complete many real-world tasks efficiently and reliably. It is, by definition, more powerful and capable than a single robot. Considerable efforts have therefore been dedicated to exploring what specific coordination mechanisms can make the most of these systems' potential. Particular contexts of interest have been cooperative sensing or perception, target tracking, or navigation for transportation tasks. This chapter concerns specifically the problem of tracking a mobile element (i.e., a *target*) with a team of robots. We describe recent research advances in control design that provide the robots with the ability to collectively perceive the target with a suitable quality, while preserving the team's ability to move with flexibility and steadiness. Such integration of desirable traits on both counts (perception and motion) is the most relevant feature of the presented contents.

In order to observe a phenomenon, it is advantageous to have different viewpoints provided by multiple sensors placed at diverse positions; this can prevent occlusions and allows to improve the quality of perception via data integration (e.g., position triangulation). With these multi-view perception capabilities one can generate a suitable model of an object that needs to be, e.g., manipulated, or transported by a robotic system. Surveillance and escorting problems can also greatly benefit from these capabilities. Motion capture systems [11] are an important multi-sensor application along these lines. However, the sensors in motion capture setups are normally static, covering an operational space that is fixed.

Considering other more general potential application scenarios, it to interesting to extend the concept of motion capture to highly dynamic targets moving across non-fixed environments; in this case, it is required to use sensors mounted on robots, which have to move to concurrently observe the target and track its motion [27]. There are multiple constraints that are essential to consider in order to solve the resulting tracking/observation problem. One type of constraints concerns the need to identify and prescribe team geometries and motion policies that optimize the quality of the collective perception of the target. There has been substantial work in this area. The approaches that have been introduced typically focus on uncertainty minimization using metrics such as mutual information or Fisher information [22, 31, 6, 29, 23, 28, 13].

However, aside from such target perception constraints, there are other equally essential requirements: these concern the suitability of the robots' motions, at both the individual and team levels. The robots' relative positions must always be such that they can perceive and/or communicate with each other, and avoid colliding with other robots or the environment. Formation control is a vast of field of research devoted to addressing these issues. It deals with controlling robot teams towards suitable arrangements called formations, defined in general by distances and/or angles [24, 7, 17, 19, 9, 30, 12, 26]. Still, standard formation controllers focus on steering the robots' relative states but do not address collective perception and target tracking objectives, as in this chapter. For flexible motion and optimal target perception, exploring alternatives beyond the scope of reaching a fixed-shape formation can be clearly advantageous. This is the approach taken in the methodology described here.

Existing works, e.g., [3, 25, 20], have already explored the use of formation-based formalisms for target observation. These motion control frameworks have lower dimensionality and/or adaptability of the robotic team's geometry compared with the method described in this chapter. One can also use strategies based on persistent motion, such as target encirclement, to solve the tracking problems we consider. Approaches along these lines include [16, 21, 8]. In these methods, however, the robots remain gyrating and thus the team does not move steadily. Steady motions –which the method described here generates– can provide increased efficiency and stability of perception data. The control policy presented in the chapter integrates the two goals mentioned above: it facilitates appropriate target perception diversity, and allows efficient control of the robots' relative states. It does so by prescribing a default target-enclosing formation, which the robots are driven towards. In particular, the robots move to reach an optimal –in terms of a global shape alignment metric– rotated version of that prescribed formation, while they track the target's motion.

Crucially, the control approach examined in this chapter allows flexibility in the formation shape –to, e.g., avoid obstacles, adapt to the size of the environment, or improve perception quality–. This is achieved by enabling each individual robot to freely select its desired distance to the target. Even with these individual adjustments, the pursued perceptual diversity is ensured: in particular, this is the case because the robots' relative positions evolve towards a pattern in which they maintain the same relative target-viewing angles as in the default formation. Via the study of the pattern rotation dynamics, we show that the behaviors of the formation and target tracking remain stable and steady, regardless of the distance selection procedure used by the robots. An additional important feature is that the method presented here applies to general 3D motions and can be implemented with local measurements, not requiring common reference frames. This independence of global coordinate references, which provides great advantages in flexibility and simplicity, is known to make team coordination harder to study and guarantee [24].

The contents of the chapter build on work presented in [4]. The chapter is characterized by an emphasis on applying the control approach that is examined to collective navigation tasks. We believe that the approach is particularly useful for these tasks [5, 18, 2, 1], which have been and remain of great importance for practical applications. The key reason is that the *target* entity considered in the proposed controller can be regarded as a *leader* for the purposes of navigation. In this context, several properties of the method reveal themselves as especially interesting, as will be discussed throughout the chapter. We use simulation tests to exemplify the usefulness of the presented control approach. In particular, results from simulation tests of a target enclosing behavior in a 3D setting and a navigation task in a 2D environment are described.

1.2 Problem formulation

We consider a team of $N > 2$ robots in $\mathbb{R}^3$ modelled as point masses. Each robot is assigned an index $i \in \{1, ..., N\}$. We assume the robots move according to a single integrator model, i.e., each of them satisfies:

$$\mathbf{\dot{q}_i} = \mathbf{u_i}, \tag{1.1}$$

where $\mathbf{q_i} \in \mathbb{R}^3$ is robot $i's$ position, and $\mathbf{u_i} \in \mathbb{R}^3$ is its control input. Further, we denote as $\mathbf{q_t} \in \mathbb{R}^3$ the position of the *target*. The robots are tasked with collectively observing this target, which is assumed to displace with finite-norm arbitrary velocity $\mathbf{v_t}$:

$$\mathbf{\dot{q}_t} = \mathbf{v_t}. \tag{1.2}$$

The positions of the robots and the target are expressed in an arbitrary global reference frame. We will refer to relative position vectors using the following notation: $\mathbf{q_{ij}} = \mathbf{q_i} - \mathbf{q_j} = -\mathbf{q_{ji}}$.

The problem addressed consists in ensuring that the target is observed from a suitable diversity of viewpoints all through its movement. Next, we describe the framework we consider. By doing so, we will be able to define the problem in more precise terms. We define a *default desired configuration (or*

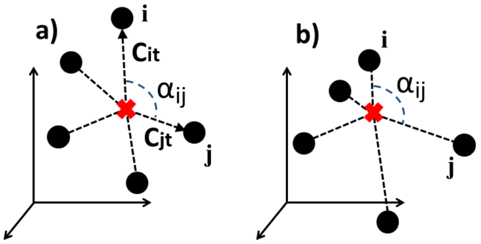

FIGURE 1.1
a) Default formation of robots (circles) which encloses a target (cross) in 3D space. In b), the formation has the same relative viewing angles α_{ij} with respect to the target as in a), with different robot-target distances. Observe that if we apply a given rotation to every vector between the target and a robot, the relative angles α_{ij} for all pairs of robots do not change.

formation), which is a reference layout of the N robots in their ambient space. This configuration is encapsulated via relative position vectors, in such a way that $\mathbf{c_{ji}} \in \mathbb{R}^3, \ \forall i, j \in \{1, ..., N\}$ denotes the default desired vector from robot i to robot j. In addition, we denote as $\mathbf{c_{ti}}$ the desired vector from robot i to the target. An objective of the task will be for the robots to *enclose* the target. In agreement with this objective, we assume that the target's desired position is right at the centroid of the desired pattern. Therefore, $\sum_{i=1}^{N} \mathbf{c_{ti}} = \mathbf{0}$. For the purposes of controller analysis, we make the assumption that the desired geometry is generic [12] and it has no exact symmetries. In reality this is not a restrictive constraint; one can modify infinitesimally any starting desired geometry to make it satisfy the constraint.

Given the presented geometric description, we can directly extract from it an angle-based configuration, encoded by the relative angles at which pairs of robots observe the target. Specifically, for any pair of robots $i, j \in \{1, ..., N\}$, their desired relative viewing angle with respect to the target is defined as $\alpha_{\mathbf{ij}} = \angle(\mathbf{c_{it}}, \mathbf{c_{jt}})$. Thus, a given desired target-observation diversity can be prescribed by defining a suitable default enclosing formation, as the one we have prescribed; such a formation directly encapsulates the desired relative target-viewing angles. We illustrate the default formation definition in Fig. 1.1.

We define the control goal as follows. The objective is for the system to reach a state where there exist a point $\mathbf{p_a} \in \mathbb{R}^3$, a rotation matrix $\mathbf{R_a} \in SO(3)$, and a set of N scalars $s_{ai} > 0$ such that:

$$\mathbf{q_i} = \mathbf{p_a} + \mathbf{R_a}\mathbf{c_{it}^{s_{ai}}}, \quad \forall i \in \{1, ..., N\}, \tag{1.3}$$

with $\mathbf{c_{it}^{s_{ai}}} = s_{ai}\mathbf{c_{it}}$, and where simultaneously $\mathbf{p_a}$ remains suitably close to $\mathbf{q_t}$.

The condition (1.3) implies the desired relative target-viewing angles are achieved with respect to $\mathbf{p_a}$, as shown next. For this fact to hold, the angle –denoted as β_{ij}– between a pair of vectors $\mathbf{q_i} - \mathbf{p_a}$ and $\mathbf{q_j} - \mathbf{p_a}$ must be equal to α_{ij}, $\forall i, j$. We rotate all such vectors by the common matrix $\mathbf{R_a}^{-1}$. This clearly does not change the angles between them. We obtain a set of vectors as follows: $\mathbf{v_i} = \mathbf{R_a}^{-1}(\mathbf{q_i} - \mathbf{p_a}) = \mathbf{c_{it}^{s_{ai}}}$ $\forall i$. Then one can see that for each i, the angle between $\mathbf{v_i}$ and $\mathbf{c_{it}}$, which we denote as β_{vci}, is zero, because $\cos(\beta_{vci}) =< \mathbf{v_i}, \mathbf{c_{it}} > /(||\mathbf{v_i}|| \cdot ||\mathbf{c_{it}}||) = s_{ai} < \mathbf{c_{it}}, \mathbf{c_{it}} > /(s_{ai}(||\mathbf{c_{it}}|| \cdot ||\mathbf{c_{it}}||)) = 1$. Thus, $\beta_{ij} = \alpha_{ij}$, $\forall i, j \in \{1, ..., N\}$.

The default desired configuration captures the relative angular constraints that define the problem we consider. Given the fact that the problem's constraints are angular, one may wonder why it can be useful or appropriate to use, as we do, a desired configuration which encodes both angles and distances. The idea is that this configuration represents a pattern that, in the absence of other task constraints, is a preferred configuration for the team to maintain. The reasons can be that the distances in that configuration are such that they facilitate the robots' interactions, and that the team's geometrical shape and size in that configuration are particularly favorable for the specific environment of operation. Nevertheless, the robots can deform the shape of the default configuration while still satisfying the relative angular constraints, as explained in subsequent sections. This capability to deform the team is very relevant as it provides great flexibility in the performance of the observation and navigation tasks addressed.

1.3 Control strategy

We describe in this section the proposed multirobot control approach. The strategy will allow to carry out the task of target enclosing for observation according to the definition put forward in the preceding section. A way to solve the addressed problem is to steer the robots towards a geometric pattern that satisfies the condition in (1.3). The default desired configuration is such a pattern. Given that a common rotation of all desired target-robot vectors leaves the angles α_{ij} unchanged, one can try to move the robots towards an *optimally* rotated version of the default formation. By doing so, the efficiency of their motions will be increased. To this end, we define the rotation by solving the problem of optimally aligning two *shapes*: the set of current positions and the set of desired points. A cost function that expresses the error of this alignment can be given by the following sum of squared distances:

$$\gamma = \sum_i \sum_j ||\mathbf{q_{ij}} - \mathbf{R_c c_{ij}}||^2, \tag{1.4}$$

where i, j both go from 1 to N, and $\mathbf{R_c} \in SO(3)$ is a rotation matrix that acts on the desired vectors. This cost function is equivalent to the one considered in orthogonal Procrustes shape alignment problems [10].

1.3.1 Rotation matrix computation

The rotation matrix is a key element of the presented control method. It enables the goals of optimality –because it minimizes γ– and independence of global coordinate systems, as will be detailed later. We can stack the position vectors between robots and obtain the following $N^2 \times 3$ matrices:

$$\begin{aligned} \mathbf{Q} &= [\mathbf{q_{11}}...\mathbf{q_{1N}}\ \mathbf{q_{21}}...\mathbf{q_{2N}}...\mathbf{q_{N1}}...\mathbf{q_{NN}}]^T \\ \mathbf{C} &= [\mathbf{c_{11}}...\mathbf{c_{1N}}\ \mathbf{c_{21}}...\mathbf{c_{2N}}...\mathbf{c_{N1}}...\mathbf{c_{NN}}]^T. \end{aligned} \tag{1.5}$$

Let us define the matrix $\mathbf{A} = \mathbf{C}^T\mathbf{Q}$ from these two sets. For optimality, we wish to find $\mathbf{R}$ such that when one chooses $\mathbf{R_c} = \mathbf{R}$ in (1.4), γ takes it minimum possible value for two given matrices $\mathbf{Q}$ and $\mathbf{C}$. It is known that this rotation can be computed using the Kabsch algorithm [14], which employs the Singular Value Decomposition (SVD) of $\mathbf{A}$, $\mathbf{A} = \mathbf{U\Lambda V}^T$, as follows:

$$\mathbf{R} = \mathbf{VDU}^T = \mathbf{V}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix}\mathbf{U}^T, \tag{1.6}$$

where $d = sign(det(\mathbf{VU}^T))$. The solution thus obtained for $\mathbf{R}$ is unique unless $rank(\mathbf{A}) < 2$ or $\mathbf{A}$ has a degenerate smallest singular value [15]. This uniqueness will be a precondition of our stability analysis presented in a subsequent section. Each robot follows the described operations to obtain at each time instant this rotation matrix, which is equal to all of them, and uses it to compute its control law.

1.3.2 Selection of desired robot-target distances

A salient aspect of the method proposed is described next. Specifically, we propose to enable each robot to select independently its desired distance to the target. This is an attractive capability: it can allow, e.g., to optimize target perception quality and to enhance navigation performance. In particular, it can be used to avoid collisions with environmental obstacles and to increase the target's comfort and safety during navigation. Thus, a robot i can choose to define a distance from the target equal to $||\mathbf{c_{ti}^s}|| = s_i||\mathbf{c_{ti}}||$, at which the robot wants to position itself. $s_i > 0$ is an upper-bounded and constant factor, which we term the *control scale* of i. This scalar is chosen freely by i and is unknown to all other robots.

1.3.3 Control law

Every robot $i \in \{1, ..., N\}$ moves according to the following closed-loop control law:

$$\mathbf{u_i} = \dot{\mathbf{q}}_\mathbf{i} = K_c(\mathbf{q_{ti}} - \mathbf{R}\mathbf{c^s_{ti}}), \tag{1.7}$$

where $K_c > 0$ is a control gain and $\mathbf{c^s_{ti}} = s_i \mathbf{c_{ti}}$ represents $i's$ desired position relative to the target, weighted by its control scale. The interpretation of this law is that the robot i moves towards a position that is situated at its desired distance to the target, and meanwhile the robot is also taking into account the team coordination goal via $\mathbf{R}$ (1.6), the optimal rotation that robot i computes at each time instant.

1.3.4 Information requirements and implementation details

In order to compute (1.7), a robot i needs to know the relative positions of the other robots relative to itself, i.e., $\mathbf{q_{ji}}$ $\forall j \neq i$. Hence, given that $\mathbf{q_{jk}} = \mathbf{q_{ji}} - \mathbf{q_{ki}}$, it is clear that i can compute the matrix $\mathbf{Q}$ (1.5) and then $\mathbf{R}$ (1.6). This relative position information required by the robot can be obtained via sensing, or by integration of the data received from other robots –through communications– with its own measurements. Note that the robots need to have global information of the team; we believe this to be a reasonable condition, given that the number of robots for a cooperative target perception task will typically be small in practical scenarios. This means that the data processing requirements will be low. It would be reasonable to consider the use of a a central unit of computation, as this would increase the efficiency of processing. The downside of this choice is that it would create a central point of failure, and make the use of communications mandatory. Robot i must also be able to perceive the target and its relative position, $\mathbf{q_{ti}}$. We do not elaborate on the issue of how the sensing of the target is performed, as this problem lies outside the scope of the chapter. Special dedicated sensors of different types could be employed for this purpose. We define the steps a robot i can follow to implement the proposed approach in Algorithm 1, where, for generality, we assume no central unit is present in the system.

The proposed control can be computed by each robot using a local and arbitrarily oriented coordinate frame. This is a prominent trait of the presented methodology, which we illustrate next for a robot k. Notice first that $\mathbf{q_{tk}}$ and $\mathbf{q_{jk}}$, $\forall j \in \{1, ..., N\}$, are all *relative* measurements; this means that there is no need for the robots to share a common coordinate origin. Moreover, we show next that the specific orientation of the reference frame of each robot is irrelevant. Let us denote as $\mathbf{P_k} \in SO(3)$ the rotation matrix between the global frame and the local frame which k uses, i.e., $\mathbf{q^P_k} = \mathbf{P_k}\mathbf{q_k}$, being $\mathbf{q^P_k}$ the position of k in a frame aligned with the local frame and with its center in the global origin. Consider (1.4), and assume the robots have fixed positions $\mathbf{q_i}$. We denote with a superscript Lk the variables that are expressed in $k's$

Algorithm 1 Robot $i's$ motion control loop

```
1. Initial Data: Template configuration ($\mathbf{c_{ji}}\ \forall j \neq i$).
2. **While** control executes **do**:
    (a) *Data collection:* Using onboard sensing/communications with other robots:
        i. Acquire relative position data of other robots and target, $\mathbf{q_{ji}}$ $\forall j \neq i$, $\mathbf{q_{ti}}$
        ii. Acquire environment constraints data
        iii. Acquire target observation data
    (b) *Control computation:* From a), compute rotation matrix $\mathbf{R}$ (1.6), determine $s_i$ using a desired distance selection algorithm, and compute control law $\mathbf{u_i}$ (1.7)
    (c) *Motion execution:* Execute physical motor commands corresponding to control $\mathbf{u_i}$
```

local frame. Let us write down the cost function (note that $\mathbf{q_{ij}^{Lk}} = \mathbf{P_k q_{ij}}$):

$$\gamma^{Lk}(\mathbf{R_c^{Lk}}) = \sum_i \sum_j ||\mathbf{q_{ij}^{Lk}} - \mathbf{R_c^{Lk} c_{ij}}||^2 =$$

$$= \sum_i \sum_j ||\mathbf{q_{ij}} - \mathbf{P_k^{-1} R_c^{Lk} c_{ij}}||^2 = \gamma(\mathbf{P_k^{-1} R_c^{Lk}}), \tag{1.8}$$

for any $\mathbf{R_c^{Lk}} \in SO(3)$. It is thus clear that the unique optimal rotations in the two frames must be such that $\mathbf{R} = \mathbf{P_k^{-1} R^{Lk}}$, i.e., $\mathbf{R^{Lk}} = \mathbf{P_k R}$. Then, since:

$$\mathbf{u_k^{Lk}} = K_c(\mathbf{P_k q_{tk}} - \mathbf{P_k R c_{tk}^s}) = \mathbf{P_k u_k}, \tag{1.9}$$

the same motion is obtained when the control law is computed in each of the two frames.

1.4 Stability analysis

We inspect the stability properties of the target enclosing strategy. Our analysis employs the following assumption:

A1: The rotation matrix (1.6) is at all times unique and differentiable as a function of the robots' positions.

Let us explain the assumption. When $\mathbf{R}$ has multiple solutions (see Section 1.3.1), this rotation is not differentiable with respect to time. Such differentiability is a prerequisite of the stability analysis we present below. The scenarios where these undesirable situations can arise, expressed by conditions on the rank and the SVD of matrix $\mathbf{A}$, are linked with perfect symmetries or singular geometries of the current and desired relative positions of the robots. These configurations have measure zero, and an infinitesimal perturbation of the robot positions brings $\mathbf{A}$ out of them. The conditions under which non-uniqueness or non-differentiability can appear correspond with specific degenerate arrangements of the robots' positions that are not attracting configurations under the proposed controller, and we disregard them via assumption A1. Further explanations on these matters are provided in Remark 1.

1.4.1 Rotation matrix dynamics

We study next the time evolution of the rotation matrix $\mathbf{R}$, which is a key element of the proposed approach.

Theorem 1. *Under the controller (1.7) and if assumption A1 is satisfied, the rotation matrix* $\mathbf{R}$ *remains constant.*

Proof. Let us first write down the inter-robot dynamics, from (1.7), as follows:

$$\dot{\mathbf{q}}_{\mathbf{ij}} = \dot{\mathbf{q}}_{\mathbf{i}} - \dot{\mathbf{q}}_{\mathbf{j}} = -K_c(\mathbf{q}_{\mathbf{ij}} - \mathbf{R}(\mathbf{c}^{\mathbf{s}}_{\mathbf{it}} - \mathbf{c}^{\mathbf{s}}_{\mathbf{jt}})). \tag{1.10}$$

We define a constant $N^2 \times 3$ matrix $\mathbf{C_s} = [\mathbf{c}^{\mathbf{s}}_{\mathbf{1t}} - \mathbf{c}^{\mathbf{s}}_{\mathbf{1t}}\ \ \mathbf{c}^{\mathbf{s}}_{\mathbf{1t}} - \mathbf{c}^{\mathbf{s}}_{\mathbf{2t}}\ \dots\ \mathbf{c}^{\mathbf{s}}_{\mathbf{Nt}} - \mathbf{c}^{\mathbf{s}}_{\mathbf{Nt}}]^T$. We can then give this expression for the dynamics of $\mathbf{Q}$ (1.5):

$$\dot{\mathbf{Q}}(t) = -K_c[\mathbf{Q} - \mathbf{C_s}\mathbf{R}^T]. \tag{1.11}$$

The dynamics of $\mathbf{A}$ is, hence, as follows:

$$\dot{\mathbf{A}} = \mathbf{C}^T\dot{\mathbf{Q}} = -K_c(\mathbf{A} - \mathbf{C}^T\mathbf{C_s}\mathbf{R}^T). \tag{1.12}$$

We will examine the evolution of $\mathbf{R}$ by studying how $\mathbf{A}$ evolves. Observe that it is clear from the definitions of $\mathbf{A}$ and $\mathbf{R}$ (Section 1.3.1) that matrix $\mathbf{AR}$ is symmetric, and therefore:

$$\mathbf{R}^T\mathbf{A}^T - \mathbf{AR} = \mathbf{0}, \tag{1.13}$$

and as a consequence:

$$\dot{\mathbf{R}}^T\mathbf{A}^T + \mathbf{R}^T\dot{\mathbf{A}}^T - \dot{\mathbf{A}}\mathbf{R} - \mathbf{A}\dot{\mathbf{R}} = \mathbf{0}. \tag{1.14}$$

One can now substitute (1.12) in this last equation and obtain:

$$\begin{aligned} &\dot{\mathbf{R}}^T\mathbf{A}^T - K_c\mathbf{R}^T\mathbf{A}^T + K_c\mathbf{R}^T\mathbf{R}\mathbf{C_s}^T\mathbf{C} \\ &+K_c\mathbf{AR} - K_c\mathbf{C}^T\mathbf{C_s}\mathbf{R}^T\mathbf{R} - \mathbf{A}\dot{\mathbf{R}} = \mathbf{0}. \end{aligned} \tag{1.15}$$

As $\mathbf{AR}$ is symmetric, it is possible to write:

$$\dot{\mathbf{R}}^T\mathbf{A}^T - \mathbf{A}\dot{\mathbf{R}} + K_c(\mathbf{C_s}^T\mathbf{C} - \mathbf{C}^T\mathbf{C_s}) = \mathbf{0}. \tag{1.16}$$

Let us now define $\mathbf{P_c} = \mathbf{C_s}^T\mathbf{C}$. We express $\mathbf{C_s} = \mathbf{C_{s1}} - \mathbf{C_{s2}}$ and, from (1.5), $\mathbf{C} = \mathbf{C_1} - \mathbf{C_2}$, where $\mathbf{C_{s1}} = [\mathbf{c^s_{1t}} \ ... \ \mathbf{c^s_{1t}} \ \mathbf{c^s_{2t}} \ ... \ \mathbf{c^s_{2t}} \ ... \ \mathbf{c^s_{Nt}}]^T$ and $\mathbf{C_{s2}} = [\mathbf{c^s_{1t}} \ ... \ \mathbf{c^s_{Nt}} \ \mathbf{c^s_{1t}} \ ... \ \mathbf{c^s_{Nt}} \ ... \ \mathbf{c^s_{Nt}}]^T$. Note that analogous expressions, without the s superscripts, apply to $\mathbf{C_1}$ and $\mathbf{C_2}$. Then, one can see that an individual element of $\mathbf{P_c}$ can be expressed as:

$$\begin{aligned}\mathbf{P_c}[i,j] = &\sum_{k=1}^{N^2} \mathbf{C_{s1}}[k,i]\mathbf{C_1}[k,j] + \mathbf{C_{s2}}[k,i]\mathbf{C_2}[k,j] \\ &- \mathbf{C_{s1}}[k,i]\mathbf{C_2}[k,j] - \mathbf{C_{s2}}[k,i]\mathbf{C_1}[k,j], \qquad (1.17)\end{aligned}$$

for $i, j = 1, 2, 3$. Thanks to the structure of the matrices, we can write: $\sum_{k=1}^{N^2} \mathbf{C_{s1}}[k,i]\mathbf{C_2}[k,j] = \sum_{k=1}^{N} \mathbf{c^s_{kt}}[i] \sum_{l=1}^{N} \mathbf{c_{lt}}[j] = 0$, because every sum along a given coordinate (x, y or z) of the N vectors from the target ($\mathbf{c_{lt}}$) is equal to zero. We reiterate that this is the case because the target lies at the centroid of the default desired configuration. In a similar fashion, one can also see that the last sum of terms in (1.17) is zero. Thus, only the two sums in the first line of (1.17) are different from zero. One then has:

$$\begin{aligned}\mathbf{P_c}[i,j] &= \sum_{k=1}^{N^2} \mathbf{C_{s1}}[k,i]\mathbf{C_1}[k,j] + \mathbf{C_{s2}}[k,i]\mathbf{C_2}[k,j] \\ &= \sum_{k=1}^{N^2} s_{p1}(k)\mathbf{C_1}[k,i]\mathbf{C_1}[k,j] + s_{p2}(k)\mathbf{C_2}[k,i]\mathbf{C_2}[k,j], \qquad (1.18)\end{aligned}$$

where $s_{p1}(k)$ and $s_{p2}(k)$ are scale values dependent only on the index k. From (1.18), one clearly sees that matrix $\mathbf{P_c}$ is symmetric. Thus, (1.16) becomes:

$$\dot{\mathbf{R}}^T\mathbf{A}^T - \mathbf{A}\dot{\mathbf{R}} = \mathbf{0}. \qquad (1.19)$$

We now refer to the analysis of an identical equation and its possible solutions given in [3, Prop. 1]. One can use the properties of the time-derivative of the rotation matrix and the SVD of $\mathbf{A}$, and consider assumption A1, to conclude that the correct solution to (1.19) is $\dot{\mathbf{R}} = \mathbf{0}$. Hence, the rotation matrix remains constant over time under the action of the proposed controller. □

Let us highlight why this result is relevant. The control scales s_i determine in a very direct manner the directions of motion –see (1.7)–. These scales are chosen freely by each robot, without coordination among them. Even though they are not used in the computation of the rotation $\mathbf{R}$, one could therefore reasonably expect that the time-evolution of $\mathbf{R}$ would be dependent on the values of these scales. However, the presented analysis has shown that $\mathbf{R}$ stays constant. A consequence of this fact is that the motion of a given robot is not influenced by the other robots' desired distances to the target. Thus, the team's motions turn out to be steady and predictable. In the context of the tracking and observation task that is addressed, these are desirable and advantageous properties.

Remark 1. *Observe from (1.12) that, as* $\mathbf{R}$ *is constant when assumption A1 holds, one can define a constant matrix* $\mathbf{A_f} = \mathbf{C}^T\mathbf{C_s}\mathbf{R}^T$ *such that:*

$$\dot{\mathbf{A}} = -K_c(\mathbf{A} - \mathbf{A_f}). \tag{1.20}$$

Hence, $\mathbf{A}$ *converges exponentially to* $\mathbf{A_f}$*. One can assume* $\mathbf{A}$ *satisfies A1 at the start of the execution. Also, given that there are no exact symmetries and alignments of the robots' positions in the desired configuration –as discussed in Section 1.2–,* $\mathbf{C}^T\mathbf{C_s}$ *satisfies A1, and the same holds for* $\mathbf{A_f}$*. Thus,* $\mathbf{A}$ *is attracted exponentially by a configuration in which A1 holds true. This provides support to the validity of the assumption that the degenerate cases can be disregarded when analyzing the controller. An alternative option would be to formulate an almost-global stability result where all possible cases would be contemplated. Regarding the geometric spaces where the robots may lie, note that the proposed method can be equally applied if the robots and target operate in 2D space. This case is explored in detail in Section 1.5.1.*□

1.4.2 Formation and tracking behaviors

Let us address the analysis of the inter-robot dynamics, and we characterize the target tracking capabilities of the proposed control approach. Let us define the *central point of observation* associated with the current positions of the robots as the following *weighted centroid*:

$$\mathbf{p_{wq}} = \frac{\sum\limits_{i=1}^{N} s_i^{-1}\mathbf{q_i}}{\sum\limits_{i=1}^{N} s_i^{-1}}. \tag{1.21}$$

It can be seen that when the positions of the robots satisfy the condition (1.3), with scales s_i, and relative to a certain point in space, then it holds that such point is equal to $\mathbf{p_{wq}}$. The dynamic behavior of this point is used in the following to study the formation and tracking dynamics.

Proposition 1. *Under the action of controller (1.7) and if A1 holds, the robots converge exponentially to a configuration where the desired relative viewing angles, and individually selected desired distances, are attained with respect to the team's weighted centroid.*

Proof. From (1.7), we express the dynamics of the relative position between two robots i and j as follows:

$$\begin{aligned}\dot{\mathbf{q}}_{\mathbf{ij}} = \dot{\mathbf{q}}_{\mathbf{i}} - \dot{\mathbf{q}}_{\mathbf{j}} &= K_c(\mathbf{q_{ti}} - \mathbf{R}\mathbf{c^s_{ti}}) - K_c(\mathbf{q_{tj}} - \mathbf{R}\mathbf{c^s_{tj}}) \\ &= -K_c(\mathbf{q_{ij}} - \mathbf{R}(s_j\mathbf{c_{tj}} - s_i\mathbf{c_{ti}})).\end{aligned} \tag{1.22}$$

Owing to Theorem 1, $\mathbf{R}$ remains constant over time, and given that the control scales are constant too, the final term of (1.22) is a constant. This allows to directly deduce the exponential convergence of each position vector $\mathbf{q_{ij}}$ to a vector $\mathbf{q_{ij}^f} = \mathbf{R}(s_j\mathbf{c_{tj}} - s_i\mathbf{c_{ti}})$. One can now define $\mathbf{p_{wq^f}} = \mathbf{q_i^f} + \mathbf{R}(s_i\mathbf{c_{ti}}) = \mathbf{q_j^f} + \mathbf{R}(s_j\mathbf{c_{tj}})$. If we isolate in these expressions $\mathbf{q_i^f}$ and substitute, for all i, in (1.21), we can find that $\mathbf{p_{wq^f}}$ is the weighted centroid of the final robot positions. Then, as $\mathbf{q_i^f} = \mathbf{p_{wq^f}} + \mathbf{R}s_i\mathbf{c_{it}}$ and all other robots satisfy an analogous expression, if one considers (1.3) and the contents in Section 1.3.2 it is possible to directly conclude the stated result. □

Proposition 2. *Under the action of the controller (1.7) and if A1 holds, the weighted centroid tracks the position of the target at all times, and the velocity of every robot becomes exponentially equal to the velocity of the weighted centroid.*

Proof. The weighted centroid's dynamics are as follows:

$$\dot{\mathbf{p}}_{\mathbf{wq}} = \frac{\sum\limits_{i=1}^{N} s_i^{-1}\dot{\mathbf{q}}_{\mathbf{i}}}{\sum\limits_{i=1}^{N} s_i^{-1}} = \frac{\sum\limits_{i=1}^{N} s_i^{-1} K_c(\mathbf{q_{ti}} - s_i\mathbf{R}\mathbf{c_{ti}})}{\sum\limits_{i=1}^{N} s_i^{-1}}$$

$$= K_c \frac{(\sum\limits_{i=1}^{N} s_i^{-1})\mathbf{q_t} - \sum\limits_{i=1}^{N} s_i^{-1}\mathbf{q_i} - \mathbf{R}\sum\limits_{i=1}^{N}\mathbf{c_{ti}}}{\sum\limits_{i=1}^{N} s_i^{-1}}. \tag{1.23}$$

Recall that $\sum\limits_{i=1}^{N}\mathbf{c_{ti}} = \mathbf{0}$ –because the target is in the centroid of the default desired pattern–. Thus, we can directly reach:

$$\dot{\mathbf{p}}_{\mathbf{wq}} = K_c(\mathbf{q_t} - \mathbf{p_{wq}}), \tag{1.24}$$

i.e., the weighted centroid tracks the target. Moreover, given that each robot's position converges exponentially to a constant position relative to the time-varying weighted centroid (Proposition 1), it follows that every one of the robots' velocities converges to that of the weighted centroid. □

This result is interesting because it implies that the central point of observation is always tracking the target's motion, even when the inter-robot formation is still far from being reached. This central point is the point that will be eventually viewed with the desired set of relative observation angles by the multirobot team. This suggests that an appropriate tracking performance will be achieved, and is a desirable type of behavior for the proposed target enclosing system.

1.5 Usage in navigation

The flexibility allowed by the method in defining each robot's motion is very handy for navigation tasks. A natural way to explore and analyze the proposed controller's application to navigation scenarios is to regard the *target* entity as a *leader* for the multirobot team. Let us elaborate on this issue next.

This leader will have a sense of global localization and environment awareness, and it will essentially direct the team's navigation towards the desired destination in the environment. In this scenario, the robots do not need to have a similar global awareness to successfully play their assigned role of tracking and perceiving the target. They can simply implement the proposed controller, which requires only local sensing capabilities, and navigate alongside the target. The target, in this case, can be considered a member of the multirobot team, rather than an external element. Therefore, the target can take into account the aim of helping the robots' navigation when making its own motion decisions. For instance, it can avoid traversing too narrow passages. The proposed control method is particularly well-suited to the described navigation scenario: its flexibility in terms of the team shape allows to adapt to a changing environment, and the steady motions it creates can increase the safety and comfort of navigation from the perspective of the target. These aspects are illustrated in a practical fashion via a simulation example (Section 1.6.2).

1.5.1 2D formulation of the controller

A case that is very relevant in practical navigation scenarios is one where the robots and the target all move in 2D (i.e., on a planar ground). It turns out that for this situation, the rotation matrix (1.6) admits a closed-form expression. In particular, the single angle α_o of the optimal rotation matrix $\mathbf{R}(\alpha_\mathbf{o}) \in SO(2)$ can be computed as follows:

$$\alpha_o = atan2(T^\perp, T), \tag{1.25}$$

where the definitions of the terms used are:

$$T = \sum_i \sum_j \mathbf{q_{ij}^T c_{ij}}, \;\; T^\perp = \sum_i \sum_j \mathbf{q_{ij}^T c_{ij}^\perp}, \tag{1.26}$$

with $\mathbf{c_{ij}^\perp} = [(0,1)^T, (-1,0)^T]\mathbf{c_{ij}}$ and the sum indices going from 1 to N.

This simpler closed-form formulation makes it easier to identify the dynamic properties of the controller. It can be shown that the fact that the rotation matrix remains constant (Theorem 1) also holds true for this 2D case. Also, from (1.25) it is clear that in this case the possible degeneracies of the controller are reduced to situations where $T = T^\perp = 0$, which are measure zero configurations.

The 2D case has peculiar features. First, it can be more appropriate to use a kinematic modeling –such as (1.1)– than in the 3D case, due to the inferior agility of the ground platforms. Also, avoidance of obstacles becomes a more pressing issue and therefore it needs to be given a greater importance when selecting the desired distances to the target (Section 1.3.2). The simulation tests provided in the section that follows will illustrate these points.

1.6 Simulation study

We describe in this section simulation tests, performed using MATLAB®, that illustrate the presented control methodology. We introduce a useful performance metric:

$$e_a = \sum_i \sum_j |\beta_{ij} - \alpha_{ij}|, \tag{1.27}$$

in which the sums are taken over all N robots. This function measures, at each time instant, the total error in the relative angles at which the robots view the weighted centroid, with respect to the desired ones.

1.6.1 Tracking of a target in 3D space

The first example includes a team of four robots, with a default configuration having the shape of a regular tetrahedron. The target's dynamics was modeled as a sum of sinusoids. Gaussian noise was added to the relative position measurements the robots used, in order to test the method's robustness. Each robot selected a different individual control scale. The values of these scales were: [1.21 0.75 0.82 1.43]. The results are illustrated in Fig. 1.2. One can see that despite the visible effects of the perturbations, the behavior is as desired, with the metric e_a being suitably regulated. The rotation matrix also remains fairly steady –up to the presence of noise–. The fact that the weighted centroid tracks the target closely can also be visualized.

1.6.2 Navigation in a 2D environment

In this example, a mobile target navigates in a 2D environment and is being enclosed and tracked by a team of six robots. The default desired configuration had the shape of a regular hexagon. As discussed in Section 1.5, in such a scenario one can exploit the adaptability of the team's shape around the target and the local character of the provided controller. The target, acting in this case as a *navigation leader*, follows a trajectory in the environment. The target has navigation capabilities that allow it to reach its destination while suitably handling the constraints posed by the environment. The robots, meanwhile, are able to operate without such global knowledge, and without a common coordinate frame of reference. In the example we describe –whose

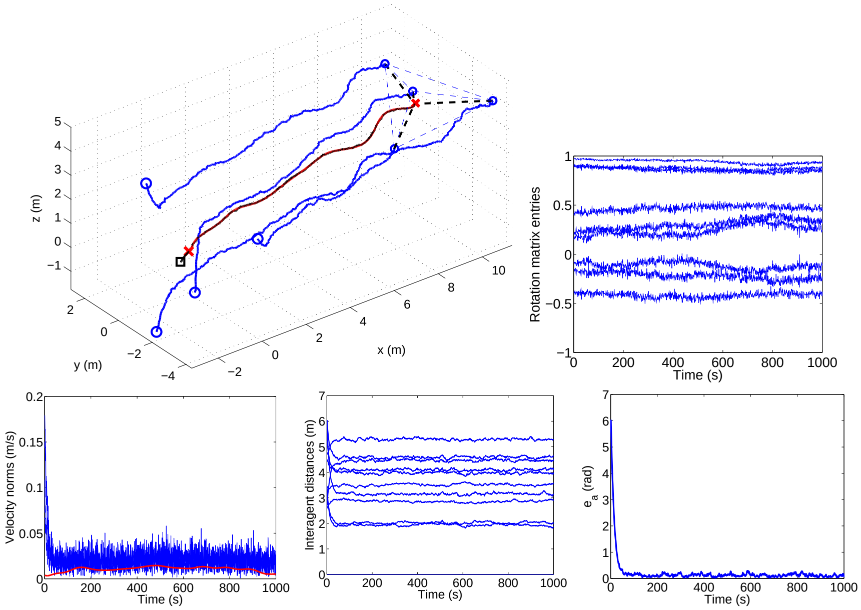

FIGURE 1.2
Results for the 3D simulation example. Top-left: paths followed by robots and target. Robots are marked as circles, target as a cross. Larger markers are used for the initial positions. Final positions are also marked. The initial weighted centroid is marked with a square and its path is shown, along with dashed lines joining it with each robot at the final configuration. Top-right: evolution of rotation matrix for one of the robots. Bottom, left to right: time evolution of velocity norms for robots (thinner lines) and target (thicker line), inter-robot distances, and error metric e_a.

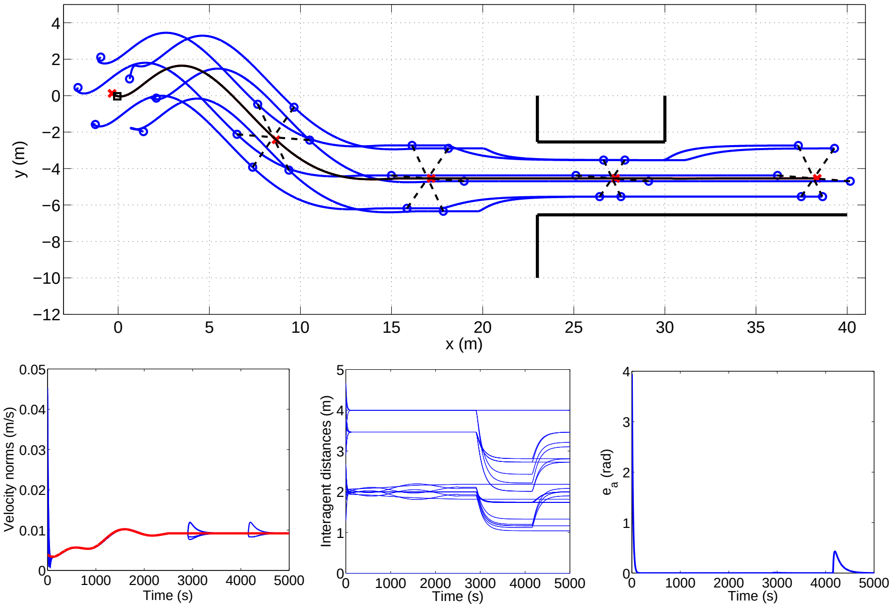

FIGURE 1.3
Results for the 2D navigation test. Top: paths followed by robots and target. Direction of motion is left to right. Robots are marked as circles, target as a cross. The initial, final and three intermediate configurations are marked. The initial weighted centroid is marked with a square and its path is shown, along with dashed lines joining it with each robot at each of the marked configurations. Environmental obstacles are marked with straight solid lines. Bottom, left to right: time evolution of velocity norms for robots (thinner lines) and target (thicker line), inter-robot distances, and error metric e_a.

results are illustrated in Fig. 1.3– the target encounters a narrow passage it has to traverse. The robots adapt their distances to the target, as required in order to avoid colliding with the environment. What is interesting to notice is that the observation quality is preserved even when these changes occur, and that a suitable performance in terms of target tracking is also retained.

1.7 Conclusion

In this chapter, we have described a novel method that a team of robots can use to enclose and observe a moving target. This motion coordination policy relies on an optimal rotation matrix and allows suitable control of the relative directions from which the different robots observe the target. Several features that make the proposed strategy interesting have been highlighted. In particular, the controller can be implemented on local coordinate frames, and it produces steady motions of the team of robots while allowing each of them to control individually its distance to the target. In this way, the approach naturally lends itself to collective navigation tasks where the robots surround the target, which acts as a team leader, while it moves across an environment. These appealing properties have been demonstrated with simulation examples.

Many improvements and additions are possible, building on the foundations laid by the presented methodology. For instance, an interesting issue to explore would be how to adapt the control framework to tasks requiring interaction with physical objects or human agents. Certain aspects, such as the robustness to disturbances and model uncertainties, can deserve a deeper formal study. It would be important to incorporate in the system model more complex and realistic robot dynamics. Finally, it would be necessary to exploit the system's degrees of freedom (i.e., the desired distances to the target) via algorithms that provide adaptivity, safety and robustness.

Bibliography

[1] L. Adouane. *Autonomous Vehicle Navigation: From Behavioral to Hybrid Multi-Controller Architectures.* CRC Press, 2016.

[2] J. Alonso-Mora, S. Baker, and D. Rus. Multi-robot navigation in formation via sequential convex programming. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4634–4641, 2015.

[3] M. Aranda, G. López-Nicolás, C. Sagüés, and M. M. Zavlanos. Three-dimensional multirobot formation control for target enclosing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 357–362, 2014.

[4] M. Aranda and Y. Mezouar. Multirobot target enclosing with freely selected observation distances. In *European Control Conference*, pages 1405–1410, 2018.

[5] N. Ayanian and V. Kumar. Decentralized feedback controllers for multiagent teams in environments with obstacles. *IEEE Transactions on Robotics*, 26(5):878–887, 2010.

[6] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Doganay, and P. N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[7] D. V. Dimarogonas and K. J. Kyriakopoulos. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica*, 44(10):2648–2654, 2008.

[8] A. Franchi, P. Stegagno, and G. Oriolo. Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance. *Autonomous Robots*, 40(2):245–265, 2016.

[9] H. García de Marina, Ming Cao, and B. Jayawardhana. Controlling rigid formations of mobile agents under inconsistent measurements. *IEEE Transactions on Robotics*, 31(1):31–39, 2015.

[10] J. C. Gower and G. B. Dijksterhuis. *Procrustes problems.* Oxford University Press, 2004.

[11] G. Guerra-Filho. Optical motion capture: Theory and implementation. *Journal of Theoretical and Applied Informatics*, 12(2):61–89, 2005.

[12] Z. Han, L. Wang, Z. Lin, and R. Zheng. Formation control with size scaling via a complex Laplacian-based approach. *IEEE Transactions on Cybernetics*, 46(10):2348–2359, 2016.

[13] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13):1660–1677, 2015.

[14] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32:922–923, 1976.

[15] K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):543–549, 1994.

[16] T.-H. Kim and T. Sugie. Cooperative control for target-capturing task based on a cyclic pursuit strategy. *Automatica*, 43(8):1426–1431, 2007.

[17] L. Krick, M. E. Broucke, and B. A. Francis. Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of Control*, 82(3):423–439, 2009.

[18] X. Li, D. Sun, and J. Yang. A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles. *Automatica*, 49(1):285–292, 2013.

[19] Z. Lin, L. Wang, Z. Han, and M. Fu. A graph Laplacian approach to coordinate-free formation stabilization for directed networks. *IEEE Trans. on Automatic Control*, 61(5):1269–1280, 2016.

[20] G. López-Nicolás, M. Aranda, and Y. Mezouar. Formation of differential-drive vehicles with field-of-view constraints for enclosing a moving target. In *IEEE International Conference on Robotics and Automation*, pages 261–266, 2017.

[21] A. J. Marasco, S. N. Givigi, and C.-A. Rabbath. Model predictive control for the dynamic encirclement of a target. In *American Control Conference*, pages 2004–2009, 2012.

[22] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[23] D. Moreno-Salinas, A. M. Pascoal, and J. Aranda. Optimal sensor placement for multiple target positioning with range-only measurements in two-dimensional scenarios. *Sensors*, 13(8):10674–10710, 2013.

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

[25] F. Poiesi and A. Cavallaro. Distributed vision-based flying cameras to film a moving target. In *IEEE/RSJ Intern. Conf. on Intell. Rob. and Systems*, pages 2453–2459, 2015.

[26] S. Ramazani, R. Selmic, and M. de Queiroz. Rigidity-based multi-agent layered formation control. *IEEE Transactions on Cybernetics*, 47(8):1902–1913, 2017.

[27] C. Robin and S. Lacroix. Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots*, 40(4):729–760, 2016.

[28] Z. Yang, X. Shi, and J. Chen. Optimal coordination of mobile sensors for target tracking under additive and multiplicative noises. *IEEE Transactions on Industrial Electronics*, 61(7):3459–3468, 2014.

[29] S. Zhao, B. M. Chen, and T. H. Lee. Optimal sensor placement for target localisation and tracking in 2D and 3D. *International Journal of Control*, 86(10):1687–1704, 2013.

[30] S. Zhao and D. Zelazo. Bearing rigidity and almost global bearing-only formation stabilization. *IEEE Transactions on Automatic Control*, 61(5):1255–1268, 2016.

[31] K. Zhou and S. I. Roumeliotis. Multirobot active target tracking with combinations of relative observations. *IEEE Transactions on Robotics*, 27 (4):678–695, 2011.