# Vision-based Distributed Formation Control without an External Positioning System

Eduardo Montijano[1,2], Eric Cristofalo[3], Dingjiang Zhou[3], Mac Schwager[4], and Carlos Sagues[2]

[1]Centro Universitario de la Defensa (CUD), Zaragoza, Spain.
[2]Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain.
[3]Department of Mechanical Engineering, Boston University, United States.
[4]Department of Aeronautics and Astronautics, Stanford University, United States.

***Abstract*—In this paper we present a fully distributed solution to drive a team of robots to reach a desired formation in the absence of an external positioning system that localizes them. Our solution addresses two fundamental problems that appear in this context. First, we propose a three dimensional distributed control law, designed at a kinematic level, that uses two simultaneous consensus controllers, one to control the relative orientations between robots, and another for the relative positions. The convergence to the desired configuration is shown by comparing the system with time-varying orientations against the equivalent approach with fixed orientations, showing that their difference vanishes as time goes to infinity. Secondly, in order to apply this controller to a group of aerial robots, we combine this idea with a novel sensor fusion algorithm to estimate the relative pose of the robots using on-board cameras and information from the Inertial Measurement Unit. The algorithm removes the influence of roll and pitch from the camera images, and estimates the relative pose between robots using a structure from motion approach. Simulation results, as well as hardware experiments with a team of three quadrotors, demonstrate the effectiveness of the controller and the vision system working together.**

## I. Introduction

Multi-robot systems have the potential to substantially impact society for the better. A wide variety of applications such as aerial manipulation [1], environmental monitoring [2], escorting [3] or entertainment [4] benefit from the cooperation of several robots. One of the challenges that multi-robot systems need to face is how to coordinate their motion without access to an external localization system. While solutions for single robots (using, e.g., SLAM techniques) are numerous, an additional challenge arises in the multi-robot context: the lack of a common reference frame. While robots can sometimes use GPS signals to get this information, in many applications it is desirable for robots to coordinate using only onboard sensing. In order to fulfill this objective, novel distributed control and perception methods based on available onboard information are required. In this paper we address the problem of controlling a group of robots to hold a specified formation using only relative pose measurements, and we also provide a vision-based algorithm to extract this relative pose information from downward facing on-board cameras mounted to the robots. We demonstrate the effectiveness of our combined vision and control system in Matlab simulations, and in hardware experiments with three quadrotor robots.

Distributed formation control has been studied extensively in the literature (see e.g., the recent survey [5]). Some approaches are based on leader-follower strategies, where each robot designs its movement depending only on a particular "leader" robot within the formation. Examples of controllers of this type have been presented using vision [6], considering relative bearing measurements [7], [8], and including obstacle avoidance [9], to name a few. On the other hand, nearest-neighbor approaches let each robot design its motion using locally available information [10] from nearby robots. Nearest neighbor approaches frequently depend on a consensus-type control algorithm, and they have the advantage that they are robust to a broad range of interaction topologies between the team of robots.

Within this last type of solutions, the formation is usually defined in terms of the relative bearings [11], [12], relative distances [13], [14], or relative positions [15]–[17] of the robots. However, a constant assumption in nearest neighbor approaches is the presence of a common rotation reference frame. In some cases it is because the robots are modeled as $d$-dimensional points without orientations whereas in others it is assumed that there exist a previous agreement about a common rotation frame, e.g., [18].

The first contribution of this paper is a three dimensional distributed consensus-based control strategy considering both relative positions and orientations for a team of first order kinematic robots. While consensus-based controllers have been deeply studied for positions [16], [17] and rotations [19]–[21] independently, there exists little research considering both elements together. The property of passivity [22] is used in [23] to show convergence to the desired 3D formation, with the orientations of all the robots eventually aligned. In [24], [25], the robots' orientations evolve in such a way that all of them reach a same value, i.e., reaching agreement on the rotation frame, and then analyze the behavior of the whole system in that consensus frame. Compared to them, in our approach

This work was funded in part by the projects DPI2012-32100, CUD2013-05 and grant CAS14/00205, and in part by NSF grants CNS-1330008 and IIS-1350904. We are grateful for this support.

the final desired rotations of the robots can be different. As a consequence, we require a different analysis, which is based on the time-varying frame described by one of the robots, instead of a fixed one. We show that, with the proposed controller, the kinematics have a structure that decouples the rotation evolution from the position, converging to the desired values. Then the position consensus can be analyzed with the different rotation matrices treated as time-varying terms that approach a known limit, finally demonstrating that the coupled positions and orientations converge to a desired relative configuration. A preliminary version of the formation control section of this article was presented in [26]. Compared to it, in this paper we extend the formation control strategy to cope with three dimensional positions and rotations.

Related with the presented controller, we also deal with the problem of estimating the relative positions and orientations of the robots. We propose to use vision sensors, as cameras are inexpensive, light, and readily available, yet they provide rich information about the environment. Furthermore, powerful distributed algorithms have recently emerged in computer vision [27], [28], which can prove to be useful in multi-robot missions [29]. Vision-based formation control with onboard cameras has already been studied in some scenarios. Usually, this problem is solved considering that the robots have a direct line of sight with some other robot. In this context we find leader-follower solutions for non-holonomic robots moving on the plane with conventional [6] and omnidirectional [30], [31] cameras. Similarly, nearest neighbor solutions for flocking and circular formations on the plane are studied with omnidirectional cameras in [32]. Leader-follower attitude-synchronization for aerial vehicles with six degrees of freedom is analyzed in [33] and bearing only formations in [8]. The need for direct observation of other robots may complicate additional tasks such as exploration or coverage. To avoid this problem, common observations of the environment are used in [34] to make the robots reach consensus in their orientations and in [35] to give a good visual coverage of the environment. While these kinds of solutions require communication of the observed features, they are better suited to solve the formation control problem in scenarios in which this problem may not represent the primary objective of the team.

The solution proposed in this paper does not require a direct line of sight between neighboring robots. Instead, we consider a situation in which the robots move parallel to a planar surface and have their cameras pointing towards this surface. This occurs for example with indoors robots moving on a plane with the cameras pointing to the ceiling or aerial vehicles with cameras pointing to the ground. Under these assumptions, we use a simple homography decomposition to obtain the relative orientation and up-to-scale relative position. We introduce an additional consensus procedure to agree upon the scale of the scene, given by the distance with respect to the plane, and show convergence to an up to scale formation configuration. Additionally, for the extra degrees of freedom of the aerial vehicles, we introduce a homography-based image rectification algorithm, which removes the effect of the robot's roll and pitch angles in the relative pose estimation using information from the on-board IMU. The extension to consider the particularities of vision sensors in the control problem has not been presented at any conference.

Summarizing, the main contributions of the paper are the following:

- An extension of the distributed formation control strategy proposed in [26] to deal with three dimensional agents. Our solution is based on consensus methods, driving the team of agents to the desired relative positions and orientations without the need of an external positioning system. We provide a rigorous theoretical study of the convergence to the objective configuration.
- A solution to the problem using vision sensors under some additional constraints. Our method combines a consensus strategy to agree upon the scene scale and a homography-based image rectification method to remove the roll and the pitch from the estimation. We prove convergence to the desired relative rotations and up to scale positions.
- Evaluation of both the controller and the pose estimation algorithm with simulations and hardware experiments with a team of three UAVs.

The remainder of the paper is organized as follows: Section II introduces the basic notation used in the paper and formally defines the formation control problem in terms of kinematic agents. Section III discusses the distributed control strategy with formal guarantees of convergence. In Section IV we propose a vision-based procedure based on homographies to estimate the relative position between pairs of robots. Simulation results are shown in Section V and hardware experiments in Section VI. Finally, Section VII presents the conclusions of the work and future lines of research.

## II. Notation and Control Problem Setup

We consider a team of $N$ rigid body agents, labeled by $\mathcal{V}$, moving in 3D space. We let $\mathcal{F}_{\mathrm{w}}$ be the fixed (unknown) world frame for the agents and $\mathcal{F}_i$ be the body frame of agent $i$. The state of the agent $i$ expressed in $\mathcal{F}_{\mathrm{w}}$ is defined by its position, $\mathbf{p}_i = [x_i, y_i, z_i]^T$, and orientation, by means of the rotation matrix $\mathbf{R}_i \in SO(3)$.

Each agent is able to move with holonomic differential kinematics with respect to its own body frame. Expressed in the world frame this is:

$$\dot{\mathbf{p}}_i = \mathbf{R}_i \mathbf{v}_i \tag{1}$$

with $\mathbf{v}_i = [v_{x_i}, v_{y_i}, v_{z_i}]^T$ the linear velocity in the x, y, and z coordinates of the body frame of agent $i$ and $\boldsymbol{w}_i = [w_{x_i}, w_{y_i}, w_{z_i}]^T$ the angular velocities along the three axis, also in the body frame of agent $i$. The time derivative of the rotation matrix, $\mathbf{R}_i$, is equal to

$$\dot{\mathbf{R}}_i = \mathbf{R}_i \mathbf{S}_i, \tag{2}$$

with $\mathbf{S}_i$ the skew-symmetric matrix associated to the angular velocity, $\boldsymbol{w}_i$,

$$\mathbf{S}_i = \begin{pmatrix} 0 & -w_{z_i} & w_{y_i} \\ w_{z_i} & 0 & -w_{x_i} \\ -w_{y_i} & w_{x_i} & 0 \end{pmatrix}. \tag{3}$$

We let $\mathbf{p} = [\mathbf{p}_1^T, \dots, \mathbf{p}_N^T]^T$ denote the concatenation of the positions of all the agents.

Given two agents, we denote by $\mathbf{p}_{ij}$ and $\mathbf{R}_{ij}$ the relative position and orientation of agent $j$ measured in the body frame of agent $i$, $\mathcal{F}_i$. The relative position between the agents is described by

$$\mathbf{p}_{ij} = [x_{ij}, y_{ij}, z_{ij}]^T = \mathbf{R}_i^T(\mathbf{p}_j - \mathbf{p}_i), \tag{4}$$

whereas the relative orientation is determined by

$$\mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j. \tag{5}$$

All these elements are graphically described in Fig. 1.

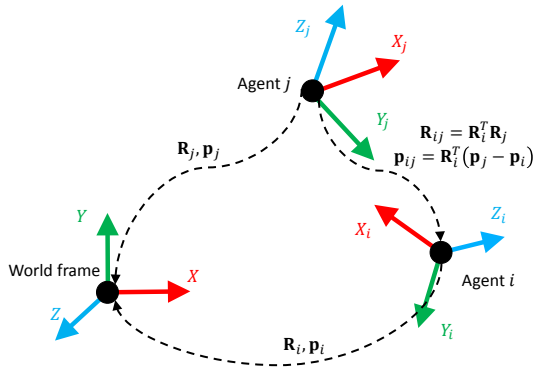



Fig. 1: Different coordinate frames.

The objective of the agents is to end up in a configuration defined by the desired relative positions and orientations, $\mathbf{p}_{ij}^*$ and $\mathbf{R}_{ij}^*$, between them.

***Assumption** 2.1: (Feasible configuration).* The desired configuration can be achieved by the team, in the sense that all the relative positions and orientations are consistent with each other. Formally speaking, it must hold that

$$\mathbf{p}_{ij}^* = \mathbf{p}_{ik}^* + \mathbf{R}_{ik}^* \mathbf{p}_{kj}^* \tag{6}$$

and

$$\mathbf{R}_{ij}^* = \mathbf{R}_{ik}^* \mathbf{R}_{kj}^*, \tag{7}$$

for all $i, j, k \in \mathcal{V}$.

Note that there are infinite sets of positions in $\mathcal{F}_\mathrm{w}$ for the agents such that they satisfy (6), all of them related by some rotation and translation. Also note that in general $\mathbf{p}_{ij} \neq -\mathbf{p}_{ji}$.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the interaction graph between the agents. The nodes in the graph, $\mathcal{V}$, are associated to the different agents, whereas the edges, $(i, j) \in \mathcal{E}$, represent the availability of the relative position and orientation of agent $j$, $[\mathbf{p}_{ij}, \mathbf{R}_{ij}]$, to agent $i$. The set of neighbors of agent $i$ is defined as the set of agents perceived by agent $i$, i.e., $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$, and $|\mathcal{N}_i|$ its cardinality. Let $\mathbf{L} = [l_{ij}]$ be the Laplacian matrix associated to $\mathcal{G}$:

$$l_{ij} = \begin{cases} |\mathcal{N}_i| & \text{if } i = j \\ -1 & \text{if } (i, j) \in \mathcal{E} \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

***Assumption** 2.2: (Connected, undirected graph).* The interaction graph, $\mathcal{G}$, is connected and undirected, i.e., there exist a path joining any two nodes and $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$.

It is well known that under this assumption $\mathbf{L}$ has one eigenvalue, $\lambda_1 = 0$, with associated right (and left) eigenvector $\mathbf{1}_N = [1, \dots, 1]^T \in \mathbb{R}^N$, and the rest of the eigenvalues are all strictly positive. Throughout the paper, we let $\ell_i$ denote the $i^{\text{th}}$ row of the Laplacian matrix. Additionally, in the paper $\mathbf{I}_k, k \in \mathbb{N}$, denotes the identity matrix of dimension $k \times k$ and the Kronecker product between two matrices is represented by $\mathbf{A} \otimes \mathbf{B}$.

## III. Distributed Formation Control with Relative Sensing

This section presents and analyzes the distributed controller followed by all the agents to reach the desired configuration. We will prove the surprising result that when each robot uses the standard consensus controller in its own frame for both position and rotation, the formation objective is reached despite the nonlinear coupling between the time varying rotations and the relative positions. The standard consensus controller for position in agent $i$'s frame is written,

$$\mathbf{v}_i = K_v \sum_{j \in \mathcal{N}_i} \left(\mathbf{p}_{ij} - \mathbf{p}_{ij}^*\right), \tag{9}$$

with $K_v > 0$ a positive gain, whereas in order to reach the desired relative rotations, the agents are driven by

$$\mathbf{w}_i = K_w \sum_{j \in \mathcal{N}_i} \log\left(\mathbf{R}_{ij}(\mathbf{R}_{ij}^*)^T\right) \tag{10}$$

with $K_w > 0$ another positive gain and log the logarithm map of a rotation matrix to transform it into a vector [21].

In the presence of a world reference frame, with all the agents aligned with the same rotation, consensus-based controllers are well known to evolve with the Laplacian matrix associated to the communication graph. However, a similar analysis in our setup is not straightforward.

In order to show that the controller (9)-(10) makes the team of agents converge to (6) we define what we call the *Relative Laplacian* matrix.

***Definition** 3.1: (Relative Laplacian Matrix).* The Relative Laplacian Matrix with respect to agent 1 is defined by

$$\mathbf{L}_1 = \mathbf{L} - \mathbf{1}_N \ell_1. \tag{11}$$

Intuitively, this matrix represents the evolution of the system in the time-varying body frame of agent 1. Using this matrix, we can remove the need of a world frame, which will let us show convergence to the desired formation by comparing the general case with a simplified version where the rotations of the agents are fixed. Also note that the choice of agent 1 is arbitrary; any other agent can be chosen to perform the analysis.

***Lemma** 3.2: (Eigenvalues of the Relative Laplacian).* Under Assumption 2.2 *(Connected, undirected graph)*, the matrices $\mathbf{L}$ and $\mathbf{L}_1$ have the same eigenvalues.

*Proof:* Since $\mathbf{L}$ is symmetric, it is diagonalizable. Thus, we can find an orthogonal matrix $\mathbf{P}$, such that $\mathbf{P}^{-1} = \mathbf{P}^T$ and

$$\mathbf{P}^T \mathbf{L} \mathbf{P} = \mathbf{\Lambda} \tag{12}$$

is a similarity transformation, with $\mathbf{\Lambda} = \text{diag}(\lambda_1 = 0, \lambda_2, \ldots, \lambda_N)$ the diagonal matrix containing all the eigenvalues of $\mathbf{L}$. Additionally, note that the columns of $\mathbf{P}$ contain the set of (normalized) eigenvectors of $\mathbf{L}$, with the special case of the first column, equal to $\mathbf{1}_N/\sqrt{N}$, i.e., $\mathbf{P} = [\mathbf{1}_N/\sqrt{N}, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_N]$, with $\boldsymbol{\gamma}_i$ the $i^{\text{th}}$ normalized eigenvector of $\mathbf{L}$. Now, if we multiply $\mathbf{L}_1$ by $\mathbf{P}^T$ and $\mathbf{P}$ on the left and right respectively, we get:

$$\mathbf{P}^T\mathbf{L}_1\mathbf{P} = \mathbf{P}^T\left(\mathbf{L} - \mathbf{1}_N\ell_1\right)\mathbf{P} = \mathbf{\Lambda} - \mathbf{P}^T\mathbf{1}_N\ell_1\mathbf{P}. \tag{13}$$

Since $\mathbf{LP} = \mathbf{P\Lambda}$, the product of $\ell_1$, which recall represents the first row of $\mathbf{L}$, by $\mathbf{P}$ is equal to the first row of $\mathbf{P\Lambda}$, which is

$$\ell_1\mathbf{P} = \left(0, \lambda_2\boldsymbol{\gamma}_{12}, \ldots, \lambda_N\boldsymbol{\gamma}_{1N}\right), \tag{14}$$

with $\boldsymbol{\gamma}_{1i}$ the first component of the $i^{\text{th}}$ eigenvector of $\mathbf{L}$. Additionally, the first eigenvector is orthogonal to all the others, which means that

$$\mathbf{P}^T\mathbf{1}_N = \left(N/\sqrt{N}, 0, \ldots, 0\right)^T. \tag{15}$$

Therefore $\mathbf{\Lambda} - \mathbf{P}^T\mathbf{1}_N\ell_1\mathbf{P}$ is an upper triangular matrix, with the same elements in the diagonal as $\mathbf{\Lambda}$. Recalling that the matrix transformation given by $\mathbf{P}^T$ and $\mathbf{P}$ preserves the eigenvalues, we conclude that the matrix $\mathbf{L}_1$ has the same eigenvalues as the matrix $\mathbf{L}$. ■

***Remark** 3.3: (Eigenvectors of the "relative" Laplacian).* Although the eigenvalues of $\mathbf{L}$ and $\mathbf{L}_1$ are the same, the eigenvectors of the two matrices are not necessarily equal. In fact, while the right eigenvector of $\mathbf{L}_1$ associated to $\lambda_1$ is the same as the one for $\mathbf{L}$, i.e., $\mathbf{1}_N$, the left eigenvector of the former matrix associated to $\lambda_1$ is $\boldsymbol{\zeta}_1 = (1, 0, \ldots, 0)^T$, which is different than $\mathbf{1}_N$. These eigenvectors will be used later in the analysis.

In the following, we focus on the convergence analysis of the controller. We first analyze a simplified scenario in which the rotations of the agents remain fixed and equal to the desired ones. After that, we use this intermediate result to prove the general case.

### *A. Analysis with fixed rotation frames*

Let us start by assuming that the agents have fixed, but not necessarily equal, rotation frames, that satisfy the following assumption:

***Assumption** 3.4: (Fixed desired orientations).* The initial orientations of the agents are such that $\mathbf{R}_{ij} = \mathbf{R}^*_{ij}$ for all $i, j \in \mathcal{V}$.

Consequently, in this scenario, $\mathbf{R}_{ij} = \mathbf{R}^*_{ij}$ and $\mathbf{w}_i = \mathbf{0}$ for all $i, j \in \mathcal{V}$.

***Proposition** 3.5: (Convergence with fixed rotation frames).* Let Assumptions 2.1 *(Feasible configuration)*, 2.2 *(Connected, undirected graph)* and 3.4 *(Fixed desired orientations)* be true. Then, using the controller in (9) the positions of the agents evolve in such a way that

$$\lim_{t\to\infty} \mathbf{p}_{ij} = \mathbf{p}^*_{ij}, \forall i, j \in \mathcal{V}. \tag{16}$$

*Proof:* For the sake of clarity, and without loss of generality, throughout the proof we consider $K_v = 1$ in eq. (9). Let us define the change of variables

$$\tilde{\mathbf{p}}_i = \mathbf{p}_{1i} - \mathbf{p}^*_{1i}. \tag{17}$$

Using (4) and (6) and Assumption 3.4

$$\mathbf{p}_{ij} = \mathbf{R}^*_{i1}(\mathbf{p}_{1j} - \mathbf{p}_{1i}),\ \mathbf{p}^*_{ij} = \mathbf{R}^*_{i1}(\mathbf{p}^*_{1j} - \mathbf{p}^*_{1i}). \tag{18}$$

Note that in the (unknown) world frame $\mathbf{R}^*_{i1} = \mathbf{R}^T_i\mathbf{R}_1$ for some fixed $\mathbf{R}_i$ and $\mathbf{R}_1$, which are also the rotation matrices that appear in (1). Thus

$$\dot{\mathbf{p}}_i = \mathbf{R}_1 \sum_{j\in\mathcal{N}_i} (\tilde{\mathbf{p}}_j - \tilde{\mathbf{p}}_i). \tag{19}$$

Differentiating (17),

$$\dot{\tilde{\mathbf{p}}}_i = \mathbf{R}^T_1(\dot{\mathbf{p}}_i - \dot{\mathbf{p}}_1) = \sum_{j\in\mathcal{N}_i} (\tilde{\mathbf{p}}_j - \tilde{\mathbf{p}}_i) - \sum_{k\in\mathcal{N}_1} (\tilde{\mathbf{p}}_k - \tilde{\mathbf{p}}_1), \tag{20}$$

which, denoting $\tilde{\mathbf{p}} = (\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_N)^T$, yields in vectorial form

$$\dot{\tilde{\mathbf{p}}} = -\left[\mathbf{L}_1 \otimes \mathbf{I}_3\right]\tilde{\mathbf{p}}. \tag{21}$$

By Lemma 3.2, the matrix $-\mathbf{L}_1$, has one eigenvalue equal to zero, and the rest of eigenvalues strictly negative. Thus, solving the differential equation (21) yields

$$\lim_{t\to\infty} \tilde{\mathbf{p}} = \left(\frac{\mathbf{1}_N\boldsymbol{\zeta}^T_1}{\boldsymbol{\zeta}^T_1\mathbf{1}_N} \otimes \mathbf{I}_3\right)\tilde{\mathbf{p}}(0) = \mathbf{1}_N\tilde{\mathbf{p}}_1(0), \tag{22}$$

with $\boldsymbol{\zeta}_1 = (1, 0, \ldots, 0)^T$. However, note that $\tilde{\mathbf{p}}_1$ is always constant and equal to $\mathbf{0}$, and therefore $\tilde{\mathbf{p}} \to \mathbf{0}$. Consequently $\mathbf{p}_{1i}$ tends to $\mathbf{p}^*_{1i}$ for all $i$, and because of Assumption 2.1 the result is proved. ■

### *B. Analysis with time-changing rotation frames*

Initially, we require to make an assumption about the initial orientations of the agents:

***Assumption** 3.6: (Relative initial orientations).* There exists a rotation matrix, $\mathbf{R}$, such that

$$\|\log\left(\mathbf{R}^T\mathbf{R}_i\mathbf{R}^*_{i1}\right)\| < \frac{\pi}{2},\ \forall i \in \mathcal{V}. \tag{23}$$

Assumption 3.6 expresses the requirement of all the initial shifted relative rotations to be contained in a ball of radius $\pi/2$ with respect to some rotation matrix. The third rotation in (23) is required to take into account the desired relative rotations but the choice of agent 1 to do the transformation is arbitrary.

The next Lemma, derived from Theorem 10 in [21], analyzes the behavior of the rotations of the agents without considering the translations.

***Lemma** 3.7: (Convergence of the rotations).* Under Assumptions 2.1 *(Feasible configuration)*, 2.2 *(Connected, undirected graph)* and 3.6 *(Relative initial orientations)*, the controller in (10) makes the rotations of all the agents to reach the desired relative values,

$$\lim_{t\to\infty} \mathbf{R}_{ij} = \mathbf{R}^*_{ij},\ \text{for all } i, j \in \mathcal{V}. \tag{24}$$

*Proof:* The demonstration follows as a corollary of Theorem 10 in [21]. The controller (10) represents a shifted version of the attitude consensus controller for relative rotations analyzed in [21]. It can be shown that using the change of variables $\mathbf{Q}_i = \mathbf{R}_i \mathbf{R}_{i1}^*$, the attitude consensus controller presented there is obtained. Assumption 3.6 represents a sufficient condition required for convergence [21] to a configuration in which $\mathbf{Q}_i^T \mathbf{Q}_j = \mathbf{I}_3$ for every pair $i$ and $j$. From there, transforming the system to the original coordinates we reach that $\mathbf{R}_{ij} \to \mathbf{R}_{ij}^*$. ■

Finally, we demonstrate convergence to the desired formation for the whole system:

***Theorem** 3.8: (Convergence with time-changing rotation frames).* Let Assumptions 2.1 *(Feasible configuration)*, 2.2 *(Connected, undirected graph)* and 3.6 *(Relative initial orientations)* be true. Then, using the controller in (9) and (10) the positions and orientations of the agents evolve in such a way that

$$\lim_{t\to\infty} \mathbf{p}_{ij} = \mathbf{p}_{ij}^*, \text{ and } \lim_{t\to\infty} \mathbf{R}_{ij} = \mathbf{R}_{ij}^*, \tag{25}$$

for all $i, j \in \mathcal{V}$.

*Proof:* For the sake of clarity, and without loss of generality, throughout the proof we consider $K_v = K_w = 1$ in eq. (9) and eq. (10). According to the kinematics in (1) and (2), using the control inputs in (9) and (10) the evolution of the orientations is decoupled from the positions of the agents. This implies that we can analyze the orientations separately and study afterwards what happens with the positions of the agents with the time-changing rotation frames.

By Lemma 3.7 we know that the rotations asymptotically converge to a constant value such that $\mathbf{R}_{ij} = \mathbf{R}_{ij}^*$ and $\mathbf{w}_i = \mathbf{0}$ for all the agents. We also know, due to Proposition 3.5, that controller (9) with constant rotations reaches the desired configuration. However, these two arguments do not guarantee that the agents will end up with the desired relative positions. A new analysis of the positions is required.

In order to simplify the notation, let us define

$$\mathbf{b}_i^* = \sum_{j\in\mathcal{N}_i} \mathbf{p}_{ij}^*, \tag{26}$$

and $\mathbf{b}^* = [(\mathbf{b}_1^*)^T, \dots, (\mathbf{b}_N^*)^T]^T$, the associated vector with all the variables.

In order to analyze the system, let us consider the change of variables

$$\mathbf{q}_i = \mathbf{p}_{1i} = \mathbf{R}_1^T(\mathbf{p}_i - \mathbf{p}_1), \tag{27}$$

that transforms all the positions relative to agent 1. Using (4), (5) and (6), the relative position $\mathbf{p}_{ij}$ can be expressed in terms of the variables $\mathbf{q}_i$,

$$\begin{aligned} \mathbf{p}_{ij} &= \mathbf{p}_{i1} + \mathbf{R}_{i1}\mathbf{p}_{1j} = -\mathbf{R}_{i1}\mathbf{p}_{1i} + \mathbf{R}_{i1}\mathbf{p}_{1j} \\ &= \mathbf{R}_i^T\mathbf{R}_1(\mathbf{p}_{1j} - \mathbf{p}_{1i}) = \mathbf{R}_i^T\mathbf{R}_1(\mathbf{q}_j - \mathbf{q}_i). \end{aligned} \tag{28}$$

Plugging (28), (26) and (9) into (1) yields the kinematics of agent $i$ in the world frame as a function of the new variables $\mathbf{q}_i$ and the constant vectors $\mathbf{b}_i^*$,

$$\dot{\mathbf{p}}_i = \mathbf{R}_i \sum_{j\in\mathcal{N}_i} \mathbf{p}_{ij} - \mathbf{R}_i\mathbf{b}_i^* = \mathbf{R}_1 \sum_{j\in\mathcal{N}_i} (\mathbf{q}_j - \mathbf{q}_i) - \mathbf{R}_i\mathbf{b}_i^*. \tag{29}$$

Now, differentiating (27) ,

$$\dot{\mathbf{q}}_i = \dot{\mathbf{R}}_1^T(\mathbf{p}_i - \mathbf{p}_1) + \mathbf{R}_1^T(\dot{\mathbf{p}}_i - \dot{\mathbf{p}}_1). \tag{30}$$

Using (2) and (4), the first term of the right hand side of (30) is equal to

$$\begin{aligned} \dot{\mathbf{R}}_1^T(\mathbf{p}_i - \mathbf{p}_1) &= (\mathbf{R}_1\mathbf{S}_1)^T(\mathbf{p}_i - \mathbf{p}_1) \\ &= \mathbf{S}_1^T\mathbf{R}_1^T(\mathbf{p}_i - \mathbf{p}_1) \\ &= \mathbf{S}_1^T\mathbf{p}_{1i} = \mathbf{S}_1^T\mathbf{q}_i. \end{aligned} \tag{31}$$

Replacing (29) in the second term yields

$$\mathbf{R}_1^T(\dot{\mathbf{p}}_i - \dot{\mathbf{p}}_1) = \sum_{j\in\mathcal{N}_i} (\mathbf{q}_j - \mathbf{q}_i) - \mathbf{R}_i 1^T \mathbf{R}_i \mathbf{b}_i^* - \sum_{k\in\mathcal{N}_1} (\mathbf{q}_k - \mathbf{q}_1) + \mathbf{b}_1^* \tag{32}$$

Therefore, combining these two terms we obtain

$$\begin{aligned} \dot{\mathbf{q}}_i = \mathbf{S}_1^T\mathbf{q}_i &+ \sum_{j\in\mathcal{N}_i} (\mathbf{q}_j - \mathbf{q}_i) - \mathbf{R}_{i1}\mathbf{b}_i^* \\ &- \sum_{k\in\mathcal{N}_1} (\mathbf{q}_k - \mathbf{q}_1) + \mathbf{b}_1^*. \end{aligned} \tag{33}$$

Let now $\mathbf{q}$ represent the vector containing all the variables, $\mathbf{q} = (\mathbf{q}_1^T, \dots, \mathbf{q}_N^T)^T$. In vectorial form, note that the second term of (33) yields the Laplacian matrix multiplied by $\mathbf{q}$, whereas the fourth term is substracting the first row of the Laplacian, $\ell_1$. Therefore, these two terms together yield the relative Laplacian, $\mathbf{L}_1$, given in Definition 3.1. Denoting $\text{diag}(\mathbf{R}_{1i}) = \text{diag}(\mathbf{R}_{11}, \dots, \mathbf{R}_{1N}) \in \mathbb{R}^{3N\times 3N}$, the block diagonal matrix defined by the $N$ relative rotational matrices of the agents with respect to agent 1, the kinematics of $\mathbf{q}$ are given by

$$\dot{\mathbf{q}} = \left[\mathbf{I}_N \otimes \mathbf{S}_1^T - \mathbf{L}_1 \otimes \mathbf{I}_3\right] \mathbf{q} - \text{diag}(\mathbf{R}_{1i})\mathbf{b}^* + \mathbf{b}_1^* \otimes \mathbf{1}_N, \tag{34}$$

where the Kronecker products are to account for the three dimensions of each variable $\mathbf{q}_i$.

In order to analyze the new system, we are going to compare it with the system described in Section III-A with fixed rotations. To distinguish them, in the proof we will denote by $\bar{\mathbf{p}}$ the system with fixed rotations. Similarly, we let $\bar{\mathbf{q}}$ be the same change of variables as in (27). Recall that with fixed rotations, we had that $\mathbf{R}_{i1} = \mathbf{R}_{i1}^*$ for all time, and therefore, the angular velocities of all the agents were zero, leading to $\mathbf{S}_1 = \mathbf{0}$. Taking into account these two elements, from (34), the kinematics of $\bar{\mathbf{q}}$ are equal to

$$\dot{\bar{\mathbf{q}}} = -\left[\mathbf{L}_1 \otimes \mathbf{I}_3\right] \bar{\mathbf{q}} - \text{diag}(\mathbf{R}_{1i}^*)\mathbf{b}^* + \mathbf{b}_1^* \otimes \mathbf{1}_N, \tag{35}$$

with $\text{diag}(\mathbf{R}_{1i}^*)$ defined as $\text{diag}(\mathbf{R}_{1i})$ with the desired relative rotations.

Let us denote

$$\mathbf{e} = \mathbf{q} - \bar{\mathbf{q}}, \tag{36}$$

the difference between the relative positions of all the agents with respect to agent 1 in the case with time-changing rotations and the case with fixed desired relative rotations. Using (34) and (35),

$$\begin{aligned} \dot{\mathbf{e}} = \left[\mathbf{I}_N \otimes \mathbf{S}_1^T - \mathbf{L}_1 \otimes \mathbf{I}_3\right] \mathbf{q} &- \text{diag}(\mathbf{R}_{1i})\mathbf{b}^* + \mathbf{b}_1^* \otimes \mathbf{1}_N \\ &+ \left[\mathbf{L}_1 \otimes \mathbf{I}_3\right] \bar{\mathbf{q}} + \text{diag}(\mathbf{R}_{1i}^*)\mathbf{b}^* - \mathbf{b}_1^* \otimes \mathbf{1}_N, \end{aligned} \tag{37}$$

which, rearranging terms, can be put in the form of a non-autonomous time-varying system

$$\dot{\mathbf{e}} = \mathbf{W}\mathbf{e} + \boldsymbol{\beta}, \tag{38}$$

with

$$\mathbf{W} = \left[\mathbf{I}_N \otimes \mathbf{S}_1^T - \mathbf{L}_1 \otimes \mathbf{I}_3\right], \tag{39}$$

and

$$\boldsymbol{\beta} = \left(\mathbf{I}_N \otimes \mathbf{S}_1^T\right) \bar{\mathbf{q}} + \left(\text{diag}(\mathbf{R}_{1i}^*) - \text{diag}(\mathbf{R}_{1i})\right) \mathbf{b}^*. \tag{40}$$

Let us remark that even when $\mathbf{e}$ depends on $\bar{\mathbf{q}}$, the evolution of this variable is known, because it was characterized in Proposition 3.5, and for that reason it can also be put as a part of $\boldsymbol{\beta}$ (the same cannot be said about $\mathbf{q}$ because its evolution is still unknown).

In order to demonstrate that $\dot{\mathbf{e}}$ converges to zero let us first consider the autonomous version of (38), i.e, $\boldsymbol{\beta} = 0$ for all time. Consider the following candidate Lyapunov function $V = \mathbf{e}^T\mathbf{e}$, which is positive definite and has time derivative equal to $\dot{V} = 2\mathbf{e}^T\mathbf{W}\mathbf{e}$. Replacing $\mathbf{W}$ by the expression in (39),

$$\begin{aligned} \dot{V} &= 2\mathbf{e}^T\mathbf{W}\mathbf{e} = 2\mathbf{e}^T\left[\mathbf{I}_N \otimes \mathbf{S}_1^T - \mathbf{L}_1 \otimes \mathbf{I}_3\right]\mathbf{e} \\ &= -2\mathbf{e}^T\left[\mathbf{L}_1 \otimes \mathbf{I}_3\right]\mathbf{e} \end{aligned} \tag{41}$$

because $\mathbf{S}_1^T$ is a skew symmetric matrix, and thus, $\mathbf{e}^T\left(\mathbf{I}_N \otimes \mathbf{S}_1^T\right)\mathbf{e} = 0$ for any value of $w_1$. Recalling Lemma (3.2),

$$-2\mathbf{e}^T\left[\mathbf{L}_1 \otimes \mathbf{I}_3\right]\mathbf{e} \leq -2\lambda_2\|\mathbf{e}\|^2, \tag{42}$$

with $\lambda_2$ the second smallest eigenvalue of $\mathbf{L}$, which because of Assumption 2.2 is strictly positive. Then, the system (38) is input-to-state stable (Lemma 4.6 in [36]).

Additionally, it holds that $\lim_{t\to\infty}\boldsymbol{\beta} = \mathbf{0}$ because we know that $\lim_{t\to\infty}\mathbf{R}_{1i} = \mathbf{R}_{1i}^*$ for all $i$ and $\lim_{t\to\infty} w_i = 0 \Rightarrow \lim_{t\to\infty}\mathbf{S}_i^T = \mathbf{0}$. Hence we have an input-to-state stable system with a bounded, vanishing disturbance. Therefore, using again [36], Lemma 4.6, we conclude that $\mathbf{e}$ converges to zero. Consequently, for all $i$ it holds that

$$\lim_{t\to\infty}\mathbf{q}_i = \lim_{t\to\infty}\bar{\mathbf{q}}_i = \mathbf{p}_{1i}^*, \tag{43}$$

and by Assumption 2.1 we conclude that all the pairs of agents also reach their desired relative positions. ■

There are some interesting issues worth mentioning about the proposed controller:

***Remark** 3.9: (Analysis in the body frame of agent* 1*).* The proof of the Theorem relies highly on the use of the relative Laplacian. The reason for this is that the final positions of the robots in the world frame are different in the case of fixed rotations and time-varying rotations. This implies that the difference of these two systems measured in the world frame is not zero, and since it is unknown, we cannot show in this frame that the desired formation has been reached. On the other hand, measuring the difference in the local body frame of agent 1, we know that in the desired configuration it should be zero. In this frame we obtain the relative Laplacian in our equations, and that is were its eigen-structure becomes useful to show the convergence.

***Remark** 3.10: (Extension to time-varying topologies).* Although we have not proven it, it should be noted that, considering that our approach is based on standard consensus-controllers, the team of agents will also reach the desired formation in a scenario with time-varying interaction topologies, given the standard requirements of bounded dwell time and periodic joint connectivity [10].

## IV. Vision-based Relative Pose Estimation and Control

Related with the problem of control we also deal with the problem of estimating the relative orientations and positions using a vision sensor. We begin with an analysis of the general 3D formulation where vision leads to a lack of absolute scale. We then consider robots moving in 3D attempting to reach a planar formation. For this objective we propose a novel algorithm for the robots to simultaneously agree on a common scale factor, reach a common altitude, and actively compensate for roll and pitch motion using IMU measurements, under the assumption that they are observing a planar scene.

In our case, instead of considering direct observations of the other robots by means of the cameras, we consider a scenario where the robots share common features of the environment and estimate their relative poses by means of structure from motion algorithms. In general, given two images with enough overlap, using these kinds of algorithms, it is possible to recover with high precision the relative rotation between the two cameras, $\mathbf{R}_{ij}$, and an up to a scale value of the relative positions, $\gamma\mathbf{p}_{ij}$, with $\gamma$ being the unknown scale. The next result demonstrates that if the scale factor, $\gamma$, is the same for all the cameras, then the formation is still reached, although with a scale factor:

***Proposition** 4.1:* Assume that the conditions in Theorem 3.8 hold and that the robots have available an accurate estimation of $\mathbf{R}_{ij}$ and a scaled estimation of the position, $\gamma\mathbf{p}_{ij}$. If $\gamma$ is equal for all the robots, then, using the controller in (9) and (10) the team reaches an up to scale formation configuration, with scale factor equal to $1/\gamma$.

*Proof:* Since the relative positions of the agents are all measured up to the constant scale factor, $\gamma$, we need to replace the term $\mathbf{p}_{ij}$ in the linear velocities in (9) by the scaled version $\gamma\mathbf{p}_{ij}$,

$$\mathbf{v}_i = K_v\sum_{j\in\mathcal{N}_i}\left(\gamma\mathbf{p}_{ij} - \mathbf{p}_{ij}^*\right) = K_v\gamma\sum_{j\in\mathcal{N}_i}\left(\mathbf{p}_{ij} - \frac{\mathbf{p}_{ij}^*}{\gamma}\right). \tag{44}$$

This is, in fact, the same consensus controller, with a different gain that does not affect convergence, and different desired relative positions. The scale factor does not affect the rotations, which means that the feasibility of the formation in eq. (6) is also satisfied for the new desired relative positions, these new ones being equal to the original ones multiplied by $1/\gamma$. This implies that all the necessary assumptions to reach the formation hold, so the team reaches the desired formation up to a scale. ■

In practice the assumption of $\gamma$ being the same for all the estimations is highly unlikely to occur. Nevertheless, in the

rest of the section we discuss a practical situation where this can be possible by means of an additional consensus procedure run to reach an agreement on $\gamma$.

We express now the rotation matrix of robot $i$ in the world frame by means of the roll, $\phi_i$, pitch, $\theta_i$, and yaw, $\psi_i$, angles, $\mathbf{R}_i = \mathbf{R}_{\psi_i}\mathbf{R}_{\theta_i}\mathbf{R}_{\phi_i}$. We assume that each robot has an identical camera aligned with its local $z$-axis and an IMU that provides a sufficiently good estimation of the global roll and pitch angles, $\phi_i$ and $\theta_i$, but not the yaw angle. The desired relative formation is now constrained by the following assumption.

***Assumption** 4.2: (Planar formation).* The desired relative formation is such that:

- The desired relative altitude between robots is zero for all pairs, i.e., $z^*_{ij} = 0$.
- The desired relative rotations are rotations only around the $z$ axis (yaw rotations).
- The roll and pitch angles are left as degrees of freedom for motion, but in the steady-state configuration they are equal to zero.

Additionally, we assume that the cameras point to a plane, which is assumed to be perpendicular to the $z$-direction of the world frame, $\mathcal{F}_\mathrm{w}$, (but not necessarily $\mathcal{F}_i$), so that its normal in $\mathcal{F}_\mathrm{w}$ is $\mathbf{n}_\mathrm{w} = [0, 0, -1]^T$. The previous assumptions find application for example when using indoor ground robots moving on a plane with the cameras pointing to the ceiling [37] or aerial vehicles with cameras pointing to the ground [38]. Let us note that although the desired formation is defined in a plane, the actual motion of the robots will not necessarily have roll and pitch angles equal to zero. For example, when using quadrotors, any motion in the local $x$ and $y$ coordinates requires that the roll and pitch angles be non-zero. Consequently, the relative position of the robots still needs to be modelled in three dimensions, even though the formation is planar.

Denote by $I_i$ and $I_j$ the images acquired by the robots, at some positions $[\mathbf{p}_i, \mathbf{R}_i]$ and $[\mathbf{p}_j, \mathbf{R}_j]$ respectively, such that the plane is visible from both of them. Under the assumption of the planar scene, both images can be related by an inter-image homography, $\mathbf{H}_{ij}$ that comprises the relative position between the robots. The homography that transforms the plane from image $j$ to image $i$ is given by

$$\mathbf{H}_{ij} = \mathbf{K}(\mathbf{R}_{ij} - \frac{\mathbf{t}_{ij}\mathbf{n}_j^T}{\delta_j})\mathbf{K}^{-1}, \tag{45}$$

where $\mathbf{K}$ is the (known) calibration matrix of the camera, which is the same for all the robots because all of them are equipped with identical cameras, $\mathbf{n}_j^T = [0, 0, 1]\mathbf{R}_j^T$, represents the normal of the plane in the body frame $\mathcal{F}_j$ and $\delta_j$ is the distance from robot $j$ to the plane. Therefore, if a particular feature in the plane appears in the pixel coordinates $\mathbf{f}_j$ in image $I_j$, and coordinates $\mathbf{f}_i$ in image $I_i$, then $\lambda\mathbf{f}_i = \mathbf{H}_{ij}\mathbf{f}_j$, with $\lambda$ the homogeneous factor.

Although it would be possible to directly use the homography $\mathbf{H}_{ij}$ to estimate $\mathbf{R}_{ij}$ and $\mathbf{t}_{ij}$ using existing algorithms [39], [40], there are some drawbacks in doing so. In general, existing approaches are known not to return a unique solution, but several (up to 4) possible transformations. Therefore the robots should first decide which one is the correct. Although the global roll and pitch are available to each robot to drive them to zero, without the knowledge of the global roll and pitch of the other robots, there is no way to eliminate the rotations caused by $\phi_{ij}$ and $\theta_{ij}$ in the general homography. Then, without additional mechanisms, the difference between the actual relative positions and the desired ones will not be accurate because of the extra relative roll and pitch in the homography. Finally, recall that we still have the objective to compute a common scale factor for the pose estimations by means of consensus, for which having all the rotations aligned along the same axis will be critical. Considering all of this, we propose instead a novel sensor fusion algorithm that combines the IMU measurements with the homography in order to solve all these issues.

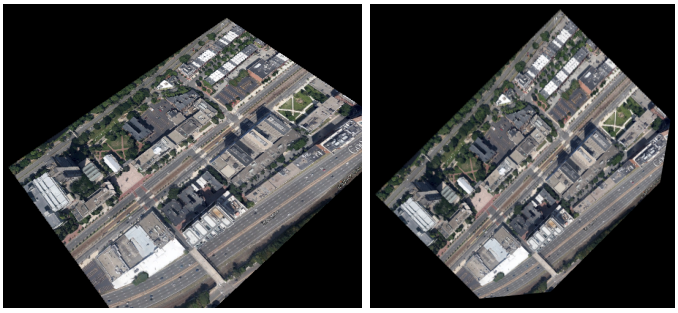


Fig. 2: Image rectification using the IMU measurements of roll and pitch. (left) Image acquired by the robot. (right) Rectified image without roll and pitch. The image is a satellite photo of the Boston University campus from Google Maps.

### A. Image rectification and relative pose estimation

The overall idea is to compute a rectification homography with the information given by the IMU, in such a way that the features are observed as if the robots were perpendicular to the ground, i.e., with roll and pitch equal to zero. In this way, the pose estimation from the homography is greatly simplified and can be used directly in the control law (9).

In order to estimate the relative pose between the robots let each one acquire an image and read the IMU at the same time. Using the information about the roll, $\phi_i$, and the pitch, $\theta_i$, angles from the IMU, we compute a rectification homography, $\mathbf{H}_i^r$, that transforms the features extracted from the image as if they were observed with the robot perpendicular to the observed plane (see Fig. 2),

$$\mathbf{H}_i^r = \mathbf{R}_{\phi_i}^{-1}\mathbf{R}_{\theta_i}^{-1}\mathbf{K}^{-1}. \tag{46}$$

The calibration matrix is post-multiplied in order to remove its effect from further computations. Therefore, given a feature, $\mathbf{f}_i$, extracted from the image, its new coordinates after the rectification will be

$$\mathbf{f}_i^r = \mathbf{H}_i^r\mathbf{f}_i. \tag{47}$$

Now, instead of exchanging the original coordinates, the robots communicate the rectified coordinates to compute the homography. Thus

$$\mathbf{f}_i^r = \mathbf{H}_{ij}^r\mathbf{f}_j^r, \tag{48}$$

which, combined with (47) yields,

$$\mathbf{H}_{ij}^r = \mathbf{H}_i^r \mathbf{H}_{ij} \left(\mathbf{H}_j^r\right)^{-1}, \tag{49}$$

where $\mathbf{H}_{ij}$ is the homography that would be obtained if no rectification was made.

Plugging eqs. (45) and (46) into (49) the homography computed between the rectified images will be of the form

$$\begin{aligned}\mathbf{H}_{ij}^r &= \mathbf{R}_{\psi_{ij}} - \frac{\mathbf{t}_{ij}[0,0,-1]^T}{z_j} \\ &= \begin{pmatrix} \cos(\psi_{ij}) & -\sin(\psi_{ij}) & -\frac{x_{ij}}{z_j} \\ \sin(\psi_{ij}) & \cos(\psi_{ij}) & -\frac{y_{ij}}{z_j} \\ 0 & 0 & 1-\frac{z_{ij}}{z_j} \end{pmatrix},\end{aligned} \tag{50}$$

where $\psi_{ij}$ denotes relative yaw angle.

From (50), the robots can compute the relative yaw and steer their rotations towards the desired relative ones,

$$w_{z_i} = K_w \sum_{j\in\mathcal{N}_i} \left( \arctan\left( \frac{[\mathbf{H}_{ij}^r]_{21}}{[\mathbf{H}_{ij}^r]_{11}} \right) - \psi_{ij}^* \right), \tag{51}$$

with $[\mathbf{H}_{ij}^r]_{ab}$ the element in row $a$ and column $b$ of $\mathbf{H}_{ij}^r$.

Additionally, $z_{ij}$ is not affected by the relative yaw and, therefore, the relative altitude can be controlled towards zero using $[\mathbf{H}_{ij}^r]_{33}$,

$$v_{z_i} = K_v \sum_{j\in\mathcal{N}_i} \left(1 - [\mathbf{H}_{ij}^r]_{33}\right). \tag{52}$$

Note also that since $z_{ij}^* = 0$, this parameter is not affected by the scale factor and can be used without additional information.

Finally, the linear velocities along the $x$ and $y$ directions are

$$\begin{pmatrix} v_{x_i} \\ v_{y_i} \end{pmatrix} = K_v \sum_{j\in\mathcal{N}_i} \left( \gamma_i(k) \begin{pmatrix} [\mathbf{H}_{ij}^r]_{13} \\ [\mathbf{H}_{ij}^r]_{23} \end{pmatrix} - \begin{pmatrix} x_{ij}^* \\ y_{ij}^* \end{pmatrix} \right), \tag{53}$$

where $\gamma_i(k)$ is the time-varying scale factor updated at discrete times.

The way the robots update their estimations of this parameter is the following. Initially, each robot assigns a positive value to the parameter, $\gamma_i(0) = \gamma_{i0} > 0$. The closer this parameter is to the distance of the robot with respect to the plane, the closer the final formation will have scale equal to one. Nevertheless, in terms of convergence, as long as $\gamma_{i0}$ is positive, the system is not affected. With the exchange of the rectified visual features, at each iteration, the robots exchange their current value $\gamma_i(k)$ with their neighbors and update it by the standard consensus protocol [41],

$$\gamma_i(k+1) = a_{ii}\gamma_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}\gamma_j(k), \tag{54}$$

where $a_{ij}$ are some weights, such that the weight matrix $\mathbf{A} = [a_{ij}]$ is doubly stochastic.

Following this simple procedure, the following result demonstrates the convergence for the whole system.

***Theorem** 4.3:* Assume that the conditions in Theorem 3.8 hold and the desired formation satisfies Assumption 4.2. Then, using the controller (51)-(54), the team of robots reaches the desired configuration up to a scale factor.

*Proof:* The rotation controller (51) is a simplification of the attitude consensus controller in $SO(3)$ applied to $SO(2)$. Considering that the rotations are not affected by the scale, the relative yaw of all the robots will converge to the desired values, see e.g. [20], [21].

Since $\mathbf{A}$ is a doubly stochastic matrix, eq. (54) denotes a classic discrete time linear consensus iteration [41], whose execution makes all the variables, $\gamma_i(k)$, converge assymptotically to the average of the initial values,

$$\lim_{k\to\infty} \gamma_i(k) = \frac{1}{N} \sum_{j\in\mathcal{V}} \gamma_{j0}, \text{ for all } i \in \mathcal{V}. \tag{55}$$

Additionally, the relative altitude evolves without being affected by the yaw rotations or the relative $x$ and $y$ positions of the robots. In fact, in this case, the evolution of $z_i$ is

$$\dot{z}_i = K_v \sum_{j\in\mathcal{N}_i} \frac{1}{z_j} \left(z_j - z_i\right), \tag{56}$$

which again corresponds to a standard consensus controller. Therefore, the altitudes of all the robots evolve in such a way that

$$\lim_{t\to\infty} z_i = \lim_{t\to\infty} z_j, \tag{57}$$

for all $i$ and $j$, and, consequently, $z_{ij} \to z_{ij}^* = 0$.

Finally, we know that for fixed desired relative rotations and a fixed scale factor, using (53), the relative positions reach the desired value. With a slightly abuse of notation, now let $\mathbf{p}_{ij} = [x_{ij}, y_{ij}]^T$. Equation (53) can also be written by

$$\mathbf{v}_i = K_v \sum_{j\in\mathcal{N}_i} \left( \frac{\gamma_i(k)}{z_j} \mathbf{p}_{ij} - \mathbf{p}_{ij}^* \right), \tag{58}$$

where both $\gamma_i(k)$ and $z_j$ converge to constant values for all the robots. Recalling that the rotations also converge to the desired value, we can use the same procedure as in the proof of Theorem 3.8 to demonstrate that the difference between the system with constant parameters and the system with time-varying ones vanishes with time, thus reaching an up to scale desired configuration. ■

***Remark** 4.4 (Convergence with image noise):* In practice, the relative pose estimation provided by the homography will be corrupted by image noise. Therefore, in a real scenario we should expect convergence to a neighborhood of the desired formation. Nevertheless, as shown in our experiments, this neighborbood is sufficiently small to consider our approach useful in real robotic applications.

## V. Simulations

In this section we test the 3D distributed controller in a simulated illustrative example.

We consider a team of $N = 6$ mobile agents and a desired formation such that the agents form an octahedron with different orientations. In particular, the four agents in the middle have their local x-axis pointing outwards from the octahedron and the top and bottom agents have it pointing

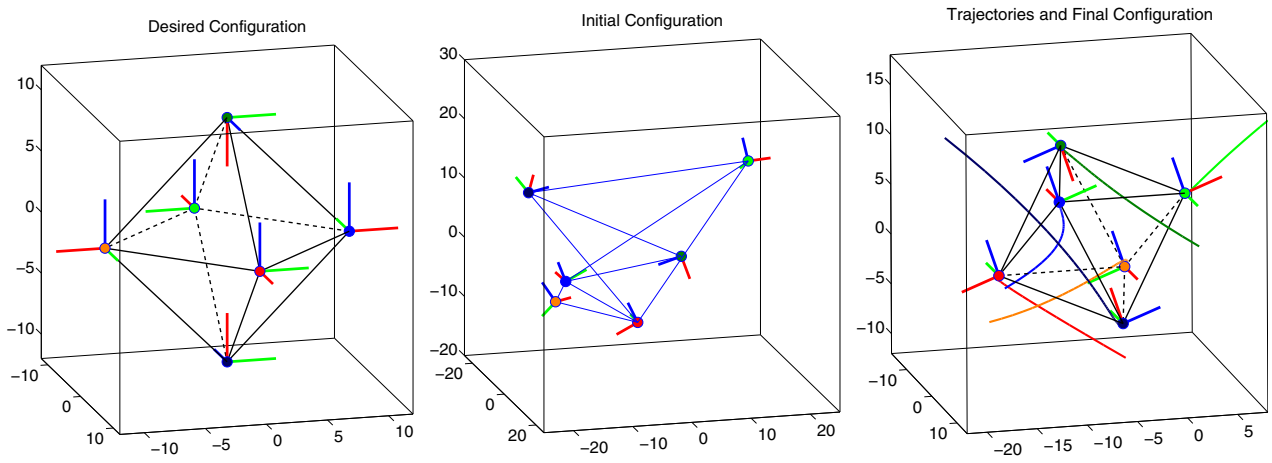



Fig. 3: 3D distributed formation control with time-varying rotation frames. The figure shows the desired configuration (left), the initial configuration (middle) and the trajectories with the final configuration of the agents (right).

inwards. This is shown in the left plot of Fig. 3, where the configuration is shown in the world frame using red, green and blue lines to represent the local x, y and z axis respectively. The black lines (solid and dashed) joining pairs of agents have been included to ease the perception of the 3D structure of the formation.

The initial configuration of the agents is depicted in the middle plot of Fig. 3, and is chosen such that the angle condition in Assumption 3.6 is satisfied. In this plot the blue lines are used to denote neighbors in the communication graph. The trajectories of the positions, as well as the final orientation of the agents obtained using the proposed controller, can be seen in the right plot of Fig. 3, where once again the black lines are used to improve the reader's perception of the 3D figure. As proven in Theorem 3.8, we can observe that the team has reached the configuration with all the relative positions and orientations equal to the desired ones, despite each one having a different frame. Note that in the left plot of Fig. 3 we are using the world frame for simplicity in depicting the desired relative positions of the agents, but, as in fact happens, the final positions of the agents in the world frame (right plot of Fig. 3) are given by a rigid body transformation of those in the left one. However, the relative positions and orientations between the agents are equal in both cases.

## VI. Hardware Experiments

The vision-based pose estimation and distributed formation controller are integrated and used to control three quadrotors into a desired formation with no global positioning information. A video of this experiment is available in the supplementary material.

### A. System Setup and Hardware

Three KMel k500 quadrotors are used in this experiment, each of them equipped with a Trendnet IP camera that provides $640 \times 400$ images (see Fig. 4). There is a rotation of $135\,\mathrm{deg}$ between the quadrotor frame and the camera frame that is

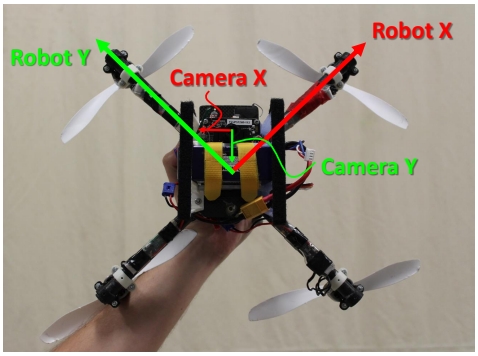



Fig. 4: Quadrotors and cameras used in the experiment.

taken into account in the relative positions estimated from the vision.

The implementation of the image processing and the computation of the kinematic inputs has been done using ROS [42] and OpenCV. The latter library includes all the necessary functions for the image processing, whereas ROS greatly simplifies communication and synchronization issues that appear when working with multiple robots. For each quadrotor there is a dedicated node in charge of processing the images of that quadrotor, extracting and publishing the SIFT descriptors [43]. Due to limitations in our hardware infrastructure, we were unable to use real IMU measurements for the image rectification. Instead, we obtained roll and pitch from the external Optitrack motion capture system, adding white gaussian noise of zero mean and standard deviation of 2 degrees to emulate a real IMU measurement. There are also three nodes (one per quadrotor) that compute homographies whenever they receive SIFT descriptors from their quadrotor or one of the other two and estimate the relative position. The homographies are computed using the 4-point DLT+RANSAC algorithm [40], to account for possible feature outliers that do not lay on the ground plane. The image processing and formation control computation together executes in a loop at 2Hz using a PC equipped with an Intel i5-4670K CPU of four cores and 16GB of RAM. The kinematic commands are sent to a different com-

puter where the low-level motion control of all the quadrotors is done using differential flatness [44]. The schematic of the system is depicted in Fig. 5.

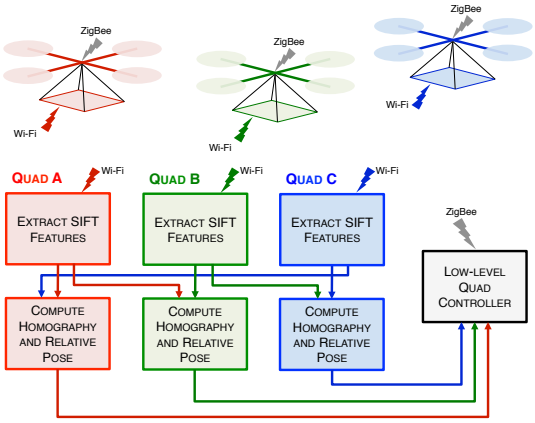



Fig. 5: Experimental system schematic. The image processing and formation control run together in a loop at 2Hz, processed in a PC equipped with an Intel i5-4670K CPU of four cores and 16GB of RAM.

The designed experiment consists in making the team of quadrotors reach three different formations. Starting from some initial conditions, the first configuration is a triangle with orientations rotated $90\,\mathrm{deg}$. The last configuration contains the three quadrotors aligned with zero relative yaw. Since the relative orientations to move from the triangle to the line do not satisfy Assumption 3.6, an intermediate formation using the same triangle but different orientations is used to allow the transition. As stated in Assumption 4.2 the desired relative roll, pitch and altitude are set to zero in the three formations in order to apply the image and scale rectification. The three desired formations are shown in Fig. 6.

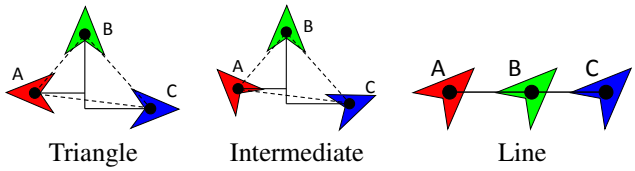



Fig. 6: Desired formations of the experiment.

In order to extract features, we have used a poster of part of the Boston University map, attached to the floor of the lab arena. An example of the kind of images acquired by the quadrotors and the matches obtained between pairs of images is given in Figure 7. The mosaic has been generated offline after the experiment for a better visualization of the matches but all the image processing required to control the quadrotors: feature extraction and matching, homography computation and rectification, relative pose estimation and kinematic control inputs, has been performed online and in real time.

### B. Experimental Results

Two external cameras have been used to record the motion of the quadrotors from outside. Figure 8 shows (from left to right) the initial position of the quadrotors and the configuration when each of the three formations is reached. For a better

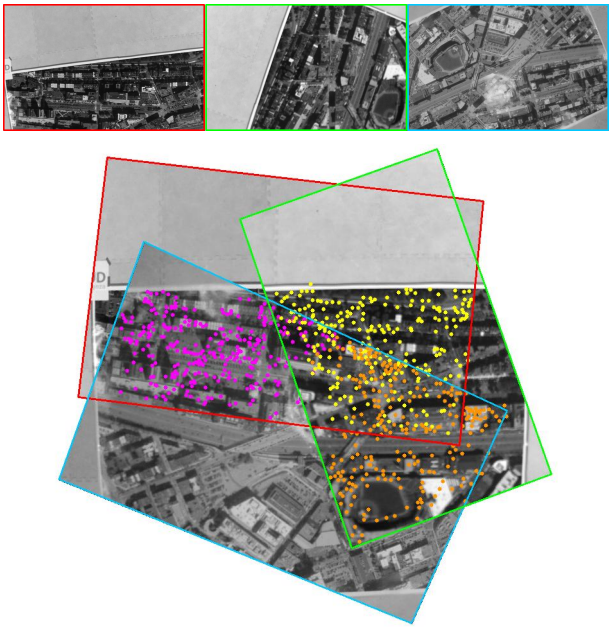


Fig. 7: Example of the images acquired by the quadrotors. (Top) Raw images from the three quadrotors. (Bottom) Homographies and matches displayed as a mosaic.

interpretation of these images, we have included the Optitrack measurements (ground truth) of the quadrotors projected into the different frames. In the top images, acquired using a GoPro camera, the red axis represents the X direction of the quadrotor frame, which coincides with the head of the arrows in Fig. 6. The green axis represents the local Y direction, whereas the blue axis is the Z direction, always pointing downwards. The color circles are used to know the role of each quadrotor in the formation (A is red, B is green and C is blue). Comparing with Fig. 6, it can be seen that the three formations are properly reached.

The relative positions and orientations estimated from the images are shown in Figure 9. These estimations (blue lines) are compared against the ground truth provided by Optitrack (red lines) and the desired relative values (black lines). It can be observed that the relative orientation estimation is very acurate, whereas the up-to-scale formation can be observed comparing the real and estimated relative positions. The constant zero estimation between quadrotors A and C in the last formation is due to the lack of enough feature matches to obtain a robust homography. Nonetheles, since both of them are able to compute their relative position with respect to quadrotor B, the whole team is still able to reach the desired configuration. It can also be observed how at the beginning of the experiment, the quadrotors have different altitudes but the homographies are able to correct them using (52). Regarding the scale factor estimation in (54), since there are only three quadrotors, all of them connected, we have just assigned a constant value of $\gamma = 1.5$ to all of them. Nonetheless, this value does not correspond with the final altitude of the quadrotors (around $1.8$ m).

Finally, the calculated linear and angular velocities are

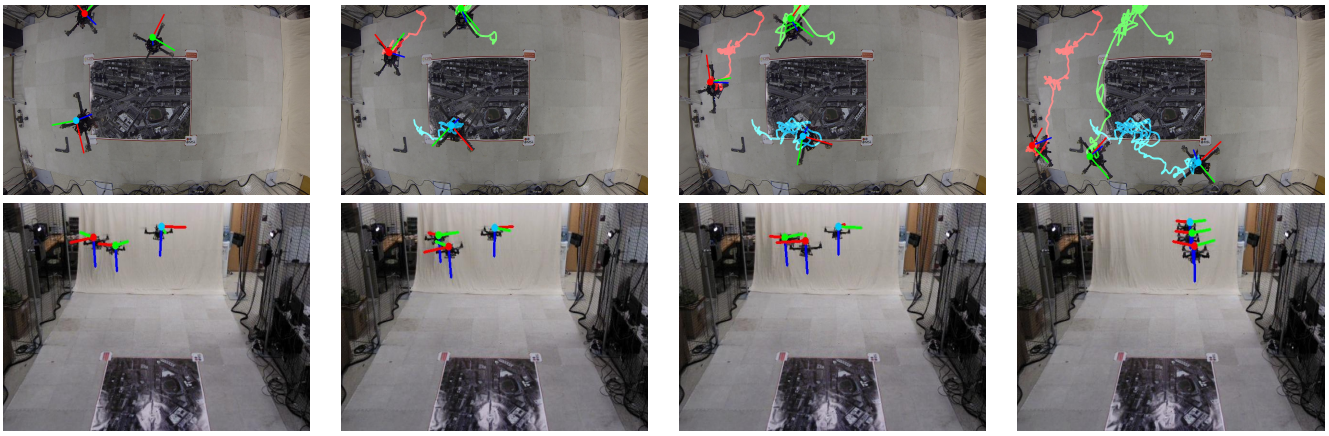



Fig. 8: Frames of the experiment. From left to right: Initial configuration of the quadrotors; triangle formation; intermediate formation; line formation. The coordinate frame axes and trajectory tails (in red, green, and blue) have been included using Optitrack measurements of the quadrotors for a better visualization of the formations.

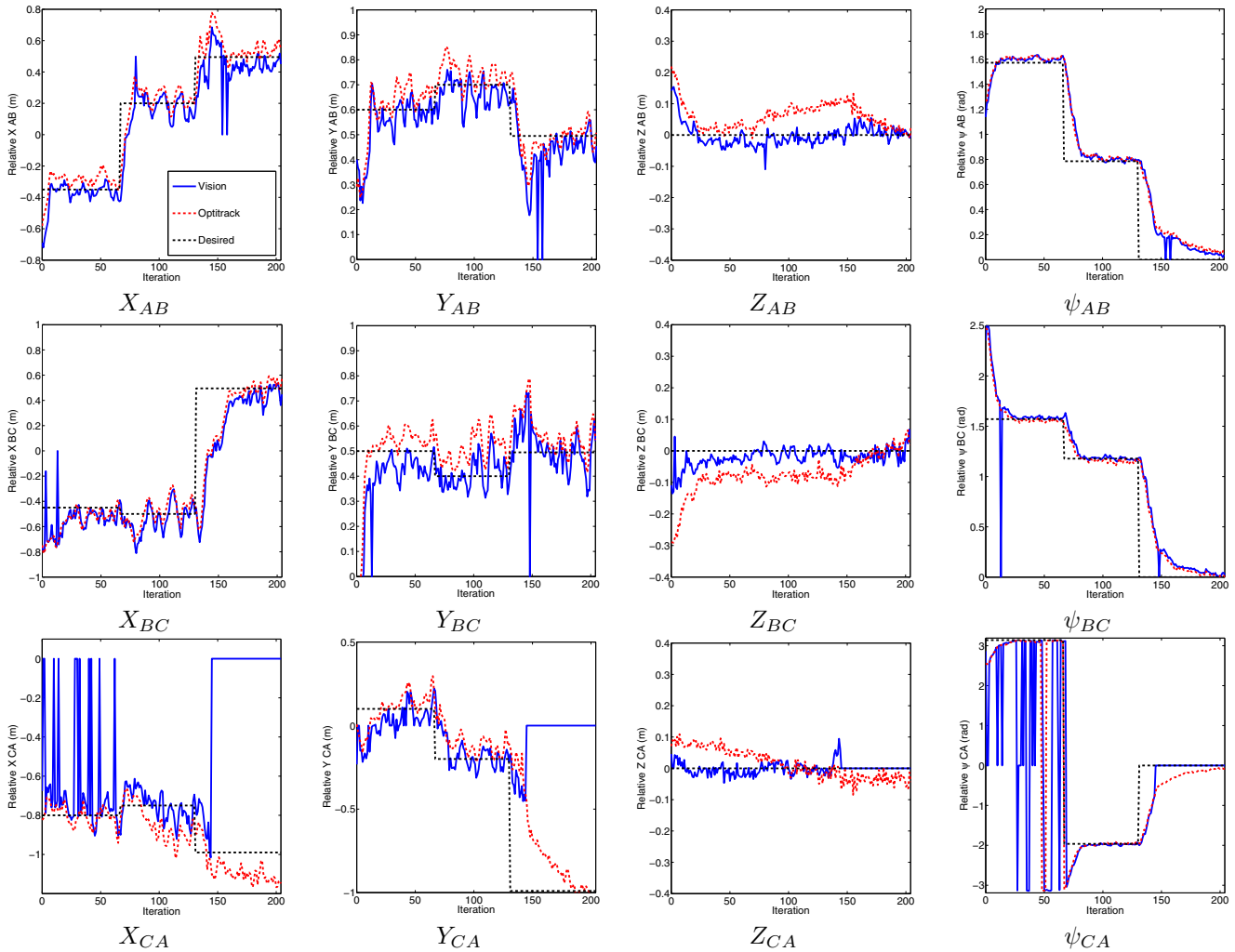



Fig. 9: Vision-based relative pose estimation between pairs of quadrotors. Blue solid line represents the estimation obtained from the rectified homographies, Red dashed line represents the real relative pose obtained from Optitrack and the dashed black line is the desired relative value in the formation. For all the quadrotors and the three formations the whole system is driven to the desired values. The up-to-scale formation can be observed comparing the real and estimated relative positions.

shown in Fig. 10. The peeks observed around iterations 65 and 130 are due to the change in the desired formation, from the triangle to the intermediate and from the intermediate to the line respectively. In the three cases velocities converge to zero as the desired formations are achieved.

## VII. Conclusions

In this paper we have presented a distributed solution to move a team of robots into a formation in the absence of an external positioning system to globally localize the robots. The solution considers two fundamental problems, the estimation of the relative positions and orientation, and the control of the robots using this information. For the latter problem we have proposed a distributed consensus controller, designed at a kinematic level, to reach a desired formation with specified relative positions and orientations in 3D. The structure of our approach has allowed us to separate the analysis in the rotations and the positions, demonstrating the convergence by considering the difference between the system with time-varying orientations and an equivalent system with fixed rotations. The relative pose estimation has been performed combining inertial measurements with computer vision algorithms in a novel image rectification scheme. The approach transforms the acquired images to look as if they were acquired perpendicular to the ground, simplifying the homography decomposition problem. We have also included a distributed consensus scheme to deal with the scale problem. Simulation results, as well as hardware experiments with three quadrotors, demonstrate the applicability of the whole solution in a real platform.

The results presented in this paper also open new interesting research possibilities. From a control perspective, in the future we are interested in studying issues of collision avoidance and role assignment for formation control without global reference frames. On the other hand, from a vision sensing perspective, a natural extension is to consider more general scenes, instead of a pure planar environment, and to integrate depth estimation methods to achieve the formation at the desired scale.

## References


[1] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, Jan. 2011.

[2] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Trans. on Automatic Control*, 55(3):650–663, March 2010.

[3] Y. Lan, Z. Lin, M. Cao, and G. Yan. A distributed reconfigurable control law for escorting and patrolling missions using teams of unicycles. In *49th IEEE Conf. on Decision and Control*, pages 5456–5461, Dec. 2010.

[4] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley. Image and animation display with multiple mobile robots. *The Int. Journal of Robotics Research*, 31(6):753–773, June 2012.

[5] K-K. Oh, M-C Park, and H-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53(3):424–440, March 2015.

[6] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. J. Taylor. Vision based formation control of multiple robots. *IEEE Trans. on Robotics and Automation*, 18(5):813–825, Oct. 2002.

[7] T. Gustavi and X. Hu. Observer-based leader-following formation control using onboard sensor information. *IEEE Trans. on Robotics*, 24(6):1457–1462, Dec. 2008.

[8] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bulfhoff, and P. R. Giordano. Modeling and control of UAV bearing formations with bilateral high-level steering. *Int. Journal of Robotics Research*, 31:1504–1525, Dec. 2012.

[9] P. Urcola and L. Montano. Cooperative robot team navigation strategies based on an environment model. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4577–4583, 2009.

[10] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules. *IEEE Trans. on Automatic Control*, 48(6):988–1001, June 2003.

[11] M. Basiri, A. N. Bishop, and P. Jensfelt. Distributed control of triangular formations with angle-only constraints. *Systems & Control Letters*, 59(1):147–154, Jan. 2010.

[12] A. Franchi and P. R. Giordano. Decentralized control of parallel rigid formations with direction constraints and bearing measurements. In *51th IEEE Int. Conf. on Decision and Control*, pages 5310–5317, Dec. 2012.

[13] M. Cao, C. Yu, and B. D.O. Anderson. Formation control using range-only measurements. *Automatica*, 47(4):776–781, April 2011.

[14] K-K Oh and H-S Ahn. Formation control of mobile agents based on inter-agent distance dynamics. *Automatica*, 47(10):2306–2312, Oct. 2011.

[15] H. Bai, M. Arcak, and J. T. Wen. Using orientation agreement to achieve planar rigid formation. In *American Control Conf.*, pages 753–758, 2008.

[16] J. Cortés. Global and robust formation-shape stabilization of relative sensing networks. *Automatica*, 45(12):2754–2762, Dec. 2009.

[17] W. Ren. Collective motion from consensus with cartesian coordinate coupling. *IEEE Trans. on Automatic Control*, 54(6):1330–1335, June 2009.

[18] A. Gasparri and M. Franceschelli. Gossip-based centroid and common reference frame estimation in multiagent systems. *IEEE Trans. on Robotics*, 30(2):524–531, April 2014.

[19] N. Mostagh and A. Jadbabaie. Distributed geodesic control laws for flocking of nonholonomic agents. *IEEE Trans. on Automatic Control*, 52(4):681–686, April 2007.

[20] R. Tron, R. Vidal, and A. Terzis. Distributed pose averaging in camera networks via consensus on SE(3). In *Second ACM/IEEE Int. Conf. on Distributed Smart Cameras*, pages 1–10, 2008.

[21] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu. Distributed attitude synchronization control of multi-agent systems with switching topologies. *Automatica*, 50(3):832–840, March 2014.

[22] T. Hatanaka, N. Chopra, M. Fujita, and M. W. Spong. *Passivity-Based Control and Estimation in Networked Robotics*. Communications and Control Engineering Series, Springer-Verlag, 2015.

[23] T. Hatanaka, Y. Igarashi, M. Fujita, and M. W. Spong. Passivity-based pose synchronization in three dimensions. *IEEE Trans. on Automatic Control*, 57(2):360–375, Feb. 2012.

[24] K-K. Oh and H-S. Ahn. Formation control and network localization via orientation alignment. *IEEE Trans. on Automatic Control*, 59(2):540–545, Feb. 2014.

[25] K-K. Oh and H-S. Ahn. Formation control of rigid bodies based on orientation alignment and position estimation. In *14th Int. Conf. on Control, Automation and Systems*, pages 740–744, Oct. 2015.

[26] E. Montijano, D. Zhou, M. Schwager, and C. Sagues. Distributed formation control without a global reference frame. In *American Control Conf.*, pages 3862–3867, 2014.

[27] B. Song, C. Ding, A. T. Kamal, J. Farrell, and A. K. Roy-Chowdhury. Distributed camera networks. *IEEE Signal Processing Magazine*, 28(3):20–31, 2011.

[28] R. Tron and R. Vidal. Distributed computer vision algorithms. *IEEE Signal Processing Magazine*, 28(3):32–45, 2011.

[29] E. Montijano and C. Sagues. Distributed multi-camera visual map-

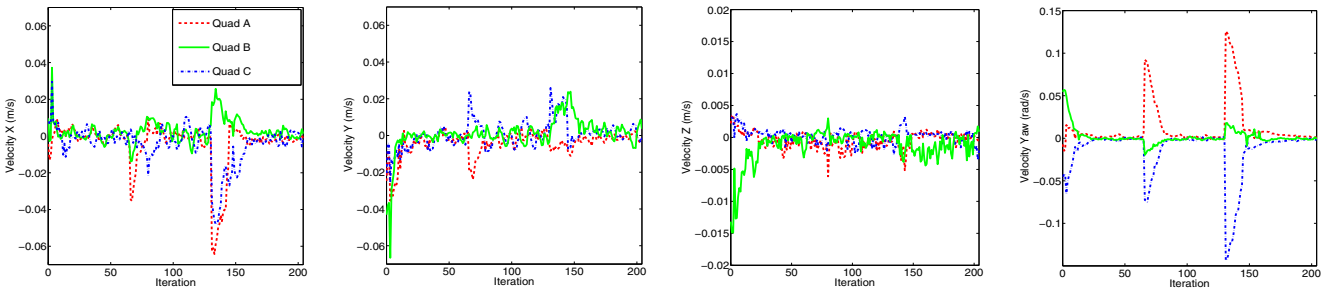



Fig. 10: Kinematic velocities computed to reach the desired formation. The change of the desired formation can be seen clearly when the angular velocity spikes (right).

ping using topological maps of planar regions. *Pattern Recognition*, 44(7):1528–1539, July 2011.

[30] R. Vidal, O. Shakernia, and S. Sastry. Following the flock [formation control]. *IEEE Robotics and Automation Magazine*, 11(4):14–20, Dec. 2004.

[31] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis. Vision-based localization for leader–follower formation control. *IEEE Trans. on Robotics*, 25(6):1431–1438, Dec. 2009.

[32] N. Mostagh, N. Michael, A. Jadbabaie, and K. Daniilidis. Vision-based, distributed control laws for motion coordination of nonholonomic robots. *IEEE Trans. on Robotics*, 25(4):851–860, Aug. 2009.

[33] T. Ibuki, T. Hatanaka, M. Fujita, and M. W. Spong. Visual feedback attitude synchronization in leader-follower type visibility structures. In *49th IEEE Conf. on Decision and Control*, pages 2486–2491, Dec. 2010.

[34] E. Montijano, J. Thunberg, X. Hu, and C. Sagues. Epipolar visual servoing for multi-robot distributed consensus. *IEEE Trans. on Robotics*, 29(5):1212–1225, Oct. 2013.

[35] M. Schwager, B. Julian, M. Angermann, and D. Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, Sept. 2011.

[36] H. K. Khalil. *Nonlinear Systems. Third Edition*. Prentice Hall, 2002.

[37] J. Courbon, Y. Mezouar, and P. Martinet. Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, 25(3):253–766, July 2008.

[38] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss. Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics and Automation Magazine*, 21(3):26–40, Sept. 2014.

[39] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[40] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3D Vision*. SpringerVerlag, 2004.

[41] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.

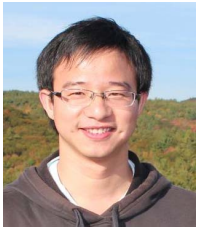
[42] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, 2009.

[43] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[44] D. Zhou and M. Schwager. Vector field following for quadrotors using differential flatness. In *IEEE Int. Conf. on Robotics and Automation*, pages 6567–6572, June 2014.

**Eduardo Montijano** (M'12) received the M.Sc. and Ph.D. degrees from the Universidad de Zaragoza, Spain, in 2008 and 2012 respectively. He has been a visiting scholar at University of California San Diego, University of California Berkeley and Boston University in the United States and at Royal Institute of Technology, in Stockholm, Sweden. He is currently a Professor at Centro Universitario de la Defensa, in Zaragoza, Spain. His main research interests include distributed algorithms, cooperative control and computer vision. His Ph.D. obtained the extraordinary award of the Universidad de Zaragoza in the 2012-2013 academic year.

**Eric Cristofalo** is a PhD candidate and National Defense Science and Engineering Graduate (NDSEG) fellow in the Department of Mechanical Engineering at Boston University in the Multi-robot Systems Lab. He received his B.S. degree in Mechanical Engineering from Drexel University in 2013. His research interests include vision-based control, active perception, and 3D reconstruction with multi-robot systems.



**Dingjiang Zhou** is a Ph.D. candidate in the Department of Mechanical Engineering Department at Boston University in the Multi-robot Systems Lab. He received his B.S. degree in 2009, and his M.S. degree in 2011, from Harbin Institute of Technology. His research interests are in UAV dynamics, agile maneuvering, trajectory planning and obstacle avoidance.

**Mac Schwager** is an assistant professor with the Aeronautics and Astronautics Department at Stanford University. He obtained his BS degree in 2000 from Stanford University, his MS degree from MIT in 2005, and his PhD degree from MIT in 2009. He was a postdoctoral researcher working jointly in the GRASP lab at the University of Pennsylvania and CSAIL at MIT from 2010 to 2012, and was an assistant professor at Boston University from 2012 to 2015. His research interests are in distributed algorithms for control, perception, and learning in groups of robots and animals. He received the NSF CAREER award in 2014.

**Carlos Sagüés** (M'00, SM'11) received the M.Sc. and Ph.D. degrees from the Universidad de Zaragoza, Spain. During the course of his Ph.D. he worked on force and infrared sensors for robots. Since 1994 he has been Associate Professor and, since 2009 Full Professor with the Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, where he has also been Head Teacher. His current research interest includes control systems, computer vision, visual robot navigation and multi-vehicle cooperative control.