Dual Quaternion Cluster-Space Formation Control

Juan I. Giribet[1], Leonardo J. Colombo[2], Patricio Moreno[3], Ignacio Mas[4], and Dimos V. Dimarogonas[5]

***Abstract*—We present a tracking controller for mobile multi-robot systems based on dual quaternion pose representations applied to formations of robots in a leader-follower configuration, by using a cluster-space state approach. The proposed controller improves system performance with respect to previous works by reducing steady-state tracking errors. The performance is evaluated through experimental field tests with a formation of an unmanned ground vehicle (UGV) and an unmanned aerial vehicle (UAV), as well as a formation of two UAVs.**

***Index Terms*—Cluster-space control, leader-follower formation, dual-quaternions, mobile robots, multi-robot systems.**

I. Introduction

Extending the concept of a single autonomous mobile robot performing a task to a group of robots has been an area of active research in last decades. One of the key elements in the operation of groups of mobile robots that require a specific spatial configuration is the control method used to coordinate the behavior of each robot [1]. Formation control strategies, where spatial constraints are defined among agents, are a powerful tool in multi-robot systems, as surveyed by [2].

A commonly used technique in the robot formation control literature is the leader-follower configuration [3]. It provides a simple and intuitive definition where a robot is designated as leader and directly follows a specified path. The rest of the robots are defined as followers and their positions are described in reference to the leader or to other follower robots that are ultimately specified with respect to the leader. This architecture lends itself to distributed schemes, as only information relative to neighboring robots is needed to define the formation. Another formation control strategy for coordination of groups of robots is the cluster-space specification introduced in [4]. This provides a simple specification and monitoring of the motion of a mobile multi-robot system allowing a suitable framework to develop formation control algorithms. This strategy is based on considering the multi-robot system as a single entity, a cluster, and specifying motions with respect to cluster attributes, such as position, orientation, and geometry. In particular, this approach allows to guide the selection of a set of independent system state variables suitable for specification, control, and monitoring. This collection of state variables constitutes the system's cluster-space and can be related to robot-specific state variables, actuator state variables, etc., through a formal set of kinematic transforms. A supervisory operator or real-time pilot specifies and monitors system motion, and centralized control computations are executed with respect to the cluster-space variables. Kinematic transforms allow compensation commands to be derived for each individual robot, and they also allow data from a variety of sensor packages to be converted to cluster-space state estimates.

Fig. 1: Multirotor UAV testbed used for formation control experiments.

Rigid body kinematics have typically been represented in terms of rotation matrices, unit quaternions, or local coordinates, such as Euler angles. Formation control strategies for multi-robot systems on the special Euclidean group have been extensively studied in the last decades and are still an active strategy in formation control for the representation of rigid body kinematics and dynamics [5]–[7]. Due to the coordinate singularities associated with local coordinate charts, it is common in mobile robots to adopt the dual quaternion representation.

In particular, in the last years, dual quaternion-based definitions that include both attitude and position of robots, as well as their associated controllers were proposed [8]–

Manuscript received: February, 24, 2020; Revised April, 20, 2021; Accepted June, 11, 2021.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments.

J. Giribet, P. Moreno and I. Mas were partially supported by PICT-2019-2371 and PICT-2019-0373 projects from Agencia Nacional de Investigaciones Científicas y Tecnológicas, and UBACyT project from the Universidad de Buenos Aires (UBA), Argentina. L. Colombo was partially supported by Ministerio de Economía, Industria y Competitividad (Spain) under grant MTM2016-76702-P, PID2019- 106715GB-C21; "Severo Ochoa Program for Centres of Excellence" (CEX2019-000904-S) and a fellowship from "La Caixa" Foundation (fellowship code LCF/BQ/PI19/11690016) and I-Link Project (Ref: linkA20079) from CSIC, Spain. D. V. Dimarogonas was supported in part by the Swedish Research Council (VR), the European Research Council (ERC), the Swedish Foundation for Strategic Research (SSF), and the Knut and Alice Wallenberg Foundation (KAW).

[1,4]Juan I. Giribet and Ignacio Mas are with Universidad de San Andrés (UdeSA) and CONICET, Argentina. {jgiribet, iamas}@conicet.gov.ar

[2]Patricio Moreno is with CONICET and UBA, Facultad de Ingeniería, Lab. de Automática y Robótica (LAR), Argentina. pmoreno@conicet.gov.ar

[3]Leonardo J. Colombo is with the Institute of Mathematical Sciences, ICMAT (CSIC-UAM-UCM-UC3M) and with Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain. leo.colombo@icmat.es

[5]D. Dimarogonas is with Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. dimos@kth.se

Digital Object Identifier (DOI): see top of this page.

[11]. This formalism also provides the most compact and computationally efficient screw algebra and lends itself to be used as a representation of the motions of rigid bodies as it simultaneously describes positions and orientations relying only on eight parameters with an underlying 6-dimensional manifold structure [12]. In the same way as homogeneous transformations, dual quaternions can describe complete body motions with one single mathematical object without representation singularities. They form one of the smallest Lie groups that represent rigid motions [13] where any sequence of rigid body motions can be represented as a sequence of dual quaternion operations, making this formalism a tool that has been proven to be useful in several applications in robotics for control and navigation [14]–[16]. Thus, some advantages of dual quaternions over other formulations are: singularity-free, shortest path interpolation, computational efficiency and compact form, and the unified representation of translations and rotations in a single invariant coordinate, among others [14], [17].

In [10] a proportional-derivative controller for body transformations using dual quaternions was proposed. The work included simulations using a single robot and the tracking control problem was developed. More recently, multi-robot formation control approaches based on dual-quaternion representations for pose consensus have been developed for robotic manipulation tasks [11]. In [18], the idea of extending dual quaternions from position and attitude representations to their use in describing position, shape, and size of a robot formation was introduced. The application of this notion to a leader-follower configuration was presented in [19]. These articles developed error-driven proportional controllers limited to tracking constant references. In this work we use dual quaternions to define the pose of the robots by relative transformations on the cluster-space variables. We design and provide theoretical guarantees together with experimental validations for a new leader-follower controller using a dual quaternion representations of the formation state. The proposed setting for cluster control advance on the state-of-the-art by extending [18] to a controller which allows to reduce steady-state errors caused by a proportional controller by adding an integral term with a forgetting factor. This also generalizes the leader-follower scheme given in [19] and improves its performance limitations.

The main contributions of this work are: (a) A new tracking control law for the pose of a single mobile robot based on a dual quaternion representation, as stated in Theorem 1. This control law compensates the steady-state tracking error. (b) Its extension to a dual quaternion representation of the leader-follower formation pose, based on cluster-space control specification, and (c) the validation of the proposed control law with simulations and experimental results.

The rest of the paper is organized as follows. The topic background is presented in Section II, where the cluster-space formulation is described, and dual quaternion errors and dynamics are introduced. In Section III, a novel controller based on dual quaternions that improves on previous works [18], [19] is proposed. This controller compensates steady-state tracking errors with an integral term and a forgetting factor, improving overall performance.

Section IV presents the cluster-state formulation, the proposed formation control approach, and convergence results. Section V shows simulations results of the controller as well as experiments on a hardware testbed, using as case examples the task of aerial escorting of a ground vehicle and a flying formation of two UAVs (see Figure 1).

II. Background

A. Cluster-Space formulation

The cluster-space approach [4], [20], considers a group of robots as a single entity, a cluster, defined by state variables that capture relevant information for the application. Consider an N-robot cluster where, without loss of generality, each robot has p degrees of freedom, then the robot-space state vector is $\boldsymbol{r} \in \mathbb{R}^m$, where $m = Np$. Let $\boldsymbol{c} \in \mathbb{R}^m$ be a state vector corresponding to the cluster variables. The appropriate selection of cluster state variables may be a function of the application, the system's design, and other criteria such as operator preferences (see [20] for details). These states are related to the robot space states through m forward kinematic transformations $\text{fwd}_k(\boldsymbol{r})$, with $k = 1, \ldots, m$. The m inverse position kinematic transformations, denoted $\text{inv}_k(\boldsymbol{c})$, relate the k-th robot-state parameter to the cluster parameters. These equations can be written as

$$\boldsymbol{c} = \mathbf{FWD}(\boldsymbol{r}) = \begin{bmatrix} \text{fwd}_1(\boldsymbol{r}) & \cdots & \text{fwd}_m(\boldsymbol{r}) \end{bmatrix}^T, \quad (1)$$

$$\boldsymbol{r} = \mathbf{INV}(\boldsymbol{c}) = \begin{bmatrix} \text{inv}_1(\boldsymbol{c}) & \cdots & \text{inv}_m(\boldsymbol{c}) \end{bmatrix}^T. \quad (2)$$

Now, let $\boldsymbol{J}(\boldsymbol{r})$ be the jacobian matrix obtained from (1), and $\boldsymbol{J}^{-1}(\boldsymbol{c})$, the one obtained from (2), the mappings between the velocities are, $\dot{\boldsymbol{c}} = \boldsymbol{J}(\boldsymbol{r})\dot{\boldsymbol{r}}$ and $\dot{\boldsymbol{r}} = \boldsymbol{J}^{-1}(\boldsymbol{c})\dot{\boldsymbol{c}}$, respectively. For a given cluster definition, configurations may exist for which the Jacobian $\boldsymbol{J}$ or its inverse $\boldsymbol{J}^{-1}$ become singular. As expected, when the formation reaches a singular pose, the system becomes unstable. We refer to [4] for a detailed discussion about singularities in the cluster-space control formulation. Following [18], we will see in Section IV the use of dual quaternions will allow us to avoid such situation. We first introduce dual quaternions to describe agents' kinematics.

B. Dual quaternions

The skew-symmetric matrix function $\boldsymbol{S}(\cdot) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ is the matrix such that $\boldsymbol{S}(\boldsymbol{v})\boldsymbol{w} = \boldsymbol{v} \times \boldsymbol{w}$, for every $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^3$, where $\times$ gives the vector product.

Let $\boldsymbol{p} \in \mathbb{R}^3$ represent the vehicle position, and let a be a frame of reference. Then $\boldsymbol{p}^a$ denotes the vehicle position expressed in frame a.

Let $\mathbb{H}$ be the set of quaternions with the standard operations [14], [17]. The set $\mathbb{H}$ can be identified with $\mathbb{R}^4$ and its operations can be written in matrix form. In fact, every $\overline{\boldsymbol{q}} \in \mathbb{H}$ can be decomposed in its vector component $\boldsymbol{q} \in \mathbb{R}^3$ and real component $q_0 \in \mathbb{R}$. Then, given $\overline{\boldsymbol{p}}, \overline{\boldsymbol{q}} \in \mathbb{H}$, with $\overline{\boldsymbol{p}} = (\boldsymbol{p}, p_0)$ and $\overline{\boldsymbol{q}} = (\boldsymbol{q}, q_0)$, the quaternion product, $\circ$, can be written as

$$\overline{\boldsymbol{p}} \circ \overline{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{p}) + \boldsymbol{I}p_0 & \boldsymbol{p} \\ -\boldsymbol{p}^T & p_0 \end{bmatrix} \begin{pmatrix} \boldsymbol{q} \\ q_0 \end{pmatrix}, \quad (3)$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. Every vector $\boldsymbol{p} \in \mathbb{R}^3$ can be identified with a quaternion $\overline{\boldsymbol{p}} \in \mathbb{H}$ by $\overline{\boldsymbol{p}} = (\boldsymbol{p}, 0)$. Given $\overline{\boldsymbol{q}} = (\boldsymbol{q}, q_0) \in \mathbb{H}$, the quaternion conjugate $\overline{\boldsymbol{q}}^* \in \mathbb{H}$, is defined as $\overline{\boldsymbol{q}}^* = (-\boldsymbol{q}, q_0)$. The quaternion norm is given by $\|\overline{\boldsymbol{q}}\|^2 = \overline{\boldsymbol{q}} \circ \overline{\boldsymbol{q}}^* = \overline{\boldsymbol{q}}^* \circ \overline{\boldsymbol{q}}$. It is well known that the unit-norm quaternion represents rotations on the real 3-sphere S^3. In fact, $\overline{\boldsymbol{q}} \in \mathbb{H}$ and $-\overline{\boldsymbol{q}}$ represent the same rotation. The unit norm quaternion and corresponding rotation matrix are related by $\boldsymbol{R}(\overline{\boldsymbol{q}}) = (q_0^2 - \boldsymbol{q}^T\boldsymbol{q})\boldsymbol{I} + 2\boldsymbol{q}\boldsymbol{q}^T + 2q_0\boldsymbol{S}(\boldsymbol{q})$.

Suppose that $\overline{\boldsymbol{q}} \in \mathbb{H}$ is a unit-norm quaternion representing the rotation between vehicle-frame b and inertial-frame i, then $\boldsymbol{p}^b$ and $\boldsymbol{p}^i$ are related as $\overline{\boldsymbol{p}}^i = \overline{\boldsymbol{q}} \circ \overline{\boldsymbol{p}}^b \circ \overline{\boldsymbol{q}}^*$. Let $\boldsymbol{\omega} = \boldsymbol{\omega}_{ib}^b(t) \in \mathbb{R}^3$ be the angular velocity of frame b with respect to frame i, in frame b. Then, the dynamics of unit quaternion $\overline{\boldsymbol{q}}$, representing the rotation from frame b to frame i is given by

$$\dot{\overline{\boldsymbol{q}}} = \frac{1}{2}\overline{\boldsymbol{q}} \circ \overline{\boldsymbol{\omega}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{S}(\boldsymbol{q}) + \boldsymbol{I}q_0 \\ -\boldsymbol{q}^T\end{bmatrix}\boldsymbol{\omega}. \tag{4}$$

A dual number is defined as $\hat{\alpha} = a + \varepsilon b$ where a and b are real numbers, called real part or principal part, and the dual part, respectively, with ε being an element having the following property: $\varepsilon \neq 0$ and $\varepsilon^2 = 0$. ε is a mathematical construct with a defined property and is not to be confused as having a small value close to 0. The same can be done to define dual quaternions, which have been proved to be useful for vehicle position and attitude representation [14], [17].

In what follows, given a unit quaternion $\overline{\boldsymbol{q}} \in \mathbb{H}$ (which represents the vehicle attitude) and $\overline{\boldsymbol{p}} \in \mathbb{H}$ with real part zero (which represents the vehicle position), dual quaternions are defined as $\boldsymbol{Q} = \overline{\boldsymbol{q}} + \varepsilon\frac{1}{2}(\overline{\boldsymbol{p}} \circ \overline{\boldsymbol{q}})$, where $\mathcal{P}(\boldsymbol{Q}) = \overline{\boldsymbol{q}}$ is the principal part and $\mathcal{D}(\boldsymbol{Q}) = \frac{1}{2}(\overline{\boldsymbol{p}} \circ \overline{\boldsymbol{q}})$ is the dual part of $\boldsymbol{Q}$. From now on, lowercase letters with upper bars are used to represent unit quaternions and uppercase letters with upper bars to represent dual quaternions.

The sum, product, and conjugation of dual quaternions can be extended from $\mathbb{H}$, taking into account that $\varepsilon^2 = 0$. The dual quaternion conjugate is given by $\boldsymbol{Q}^* = \mathcal{P}(\boldsymbol{Q})^* - \varepsilon\mathcal{D}(\boldsymbol{Q})^*$.

Observe that given $\boldsymbol{Q}$, it is possible to recover vehicle attitude and position as follows: $\overline{\boldsymbol{q}} = \mathcal{P}(\boldsymbol{Q})$, and $\overline{\boldsymbol{p}} = 2\mathcal{D}(\boldsymbol{Q}) \circ \mathcal{P}(\boldsymbol{Q})^*$. The time derivative of $\boldsymbol{Q}$ can be obtained with the derivatives of the principal and dual parts. In fact,

$$\dot{\mathcal{P}}(\boldsymbol{Q}) = \dot{\overline{\boldsymbol{q}}} = \frac{1}{2}\overline{\boldsymbol{q}} \circ \overline{\boldsymbol{\omega}},\ 2\dot{\mathcal{D}}(\boldsymbol{Q}) = \mathcal{D}(\boldsymbol{Q}) \circ \overline{\boldsymbol{\omega}} + \overline{\boldsymbol{v}} \circ \mathcal{P}(\boldsymbol{Q}) \tag{5}$$

where the notation $\overline{\boldsymbol{v}} = \dot{\overline{\boldsymbol{p}}}$ was introduced and the last equality follows from equation (4). The time evolution of $\boldsymbol{Q}$, hence vehicle pose, is given by the commanded angular velocity $\boldsymbol{\omega}$ (in vehicle frame) and linear velocity $\boldsymbol{v}$ (in inertial frame).

C. Dual quaternions error dynamic

We wish to design a control law that improves performances in [18] by acting on the steady-state tracking errors by incorporating an integral term with a forgetting factor to the compensation signal to achieve pose stabilization, based on dual quaternion representations, to be suitable for the multi-robot cluster-space representation in Section IV. To do that, first it is needed to compute the corresponding expression for the error dynamics.

Let $\boldsymbol{Q}$ be the dual quaternion representing the current attitude and position of vehicle and let $\boldsymbol{Q}_d$ be the desired dual quaternion, i.e., the desired vehicle pose. The dual quaternion error is defined as $\delta\boldsymbol{Q} = \boldsymbol{Q}_d^* \circ \boldsymbol{Q} = \overline{\boldsymbol{\delta q}} + \varepsilon\frac{1}{2}\overline{\boldsymbol{\delta p}}^b \circ \overline{\boldsymbol{\delta q}}$, where $\overline{\boldsymbol{\delta q}} = \overline{\boldsymbol{q}}_d^* \circ \overline{\boldsymbol{q}} = (\boldsymbol{\delta q}, \delta q_0)$ and $\overline{\boldsymbol{\delta p}} = \overline{\boldsymbol{p}} - \overline{\boldsymbol{p}}_d = (\boldsymbol{\delta p}, 0)$. The term $\overline{\boldsymbol{\delta p}}^b = \overline{\boldsymbol{q}}_d^* \circ \overline{\boldsymbol{\delta p}} \circ \overline{\boldsymbol{q}}_d = (\boldsymbol{\delta p}^b, 0)$ can be interpreted as the position error respect to the desired vehicle frame. Observe that $\mathcal{P}(\delta\boldsymbol{Q}) = \overline{\boldsymbol{\delta q}}$ and $\mathcal{D}(\delta\boldsymbol{Q}) = \frac{1}{2}(\overline{\boldsymbol{\delta p}}^b \circ \overline{\boldsymbol{\delta q}})$. In order to simplify notation, $\overline{\boldsymbol{d}} := (\boldsymbol{d}, d_0) = \mathcal{D}(\delta\boldsymbol{Q})$ is used.

Lemma 1. *Let $\overline{\boldsymbol{\omega}} = (\boldsymbol{\omega}, 0)$, $\overline{\boldsymbol{v}} = (\boldsymbol{v}, 0) \in \mathbb{H}$, $\overline{\boldsymbol{\omega}}_d = (\boldsymbol{\omega}_d, 0)$, $\overline{\boldsymbol{v}}_d = (\boldsymbol{v}_d, 0) \in \mathbb{H}$, such that equation (5) is satisfied for $\boldsymbol{Q}$ and $\boldsymbol{Q}_d$, respectively. Let $\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_d$ and $\overline{\boldsymbol{\delta v}}^b = (\boldsymbol{\delta v}^b, 0) = \overline{\boldsymbol{q}}_d^* \circ \dot{\overline{\boldsymbol{\delta p}}} \circ \overline{\boldsymbol{q}}_d$. Then, the error $\delta\boldsymbol{Q}$ satisfies*

$$\dot{\delta\boldsymbol{Q}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta q}) + \boldsymbol{I}\delta q_0 & 0 \\ -\boldsymbol{\delta q}^T & 0 \\ \boldsymbol{S}(\boldsymbol{d}) + \boldsymbol{I}d_0 & -\boldsymbol{S}(\boldsymbol{\delta q}) + \boldsymbol{I}\delta q_0 \\ -\boldsymbol{d}^T & -\boldsymbol{\delta q}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta\omega} \\ \boldsymbol{\delta v}^b\end{bmatrix} + \begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta q}) \\ 0 \\ \boldsymbol{S}(\boldsymbol{d}) \\ 0\end{bmatrix}\boldsymbol{\omega}_d,$$

Proof. By equation (4) it follows that

$$\dot{\mathcal{P}}(\delta\boldsymbol{Q}) = \dot{\overline{\boldsymbol{\delta q}}} = \dot{\overline{\boldsymbol{q}}}_d^{\,*} \circ \overline{\boldsymbol{q}} + \overline{\boldsymbol{q}}_d^* \circ \dot{\overline{\boldsymbol{q}}} = \frac{1}{2}\left(-\overline{\boldsymbol{\omega_d}} \circ \overline{\boldsymbol{\delta q}} + \overline{\boldsymbol{\delta q}} \circ \overline{\boldsymbol{\omega}}\right)$$

Taking $\boldsymbol{\delta\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_d$, and applying equation (3),

$$\dot{\overline{\boldsymbol{\delta q}}} = \frac{1}{2}\overline{\boldsymbol{\delta q}} \circ \overline{\boldsymbol{\delta\omega}} + \begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta q})\boldsymbol{\omega}_d \\ 0\end{bmatrix}. \tag{6}$$

In order to calculate the derivative of the dual part of $\delta\boldsymbol{Q}$, notice that $\dot{\boldsymbol{\delta p}}^b = \boldsymbol{S}(\boldsymbol{\delta p}^b)\boldsymbol{\omega}_d + \boldsymbol{\delta v}^b$. Then

$$\begin{aligned}\dot{\overline{\boldsymbol{d}}} := \dot{\mathcal{D}}(\delta\boldsymbol{Q}) &= \frac{1}{2}\left(\dot{\overline{\boldsymbol{\delta p}}}^b \circ \overline{\boldsymbol{\delta q}} + \frac{1}{2}\overline{\boldsymbol{\delta p}}^b \circ \overline{\boldsymbol{\delta q}} \circ \overline{\boldsymbol{\delta\omega}} + \overline{\boldsymbol{\delta p}}^b \circ \begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta q})\boldsymbol{\omega}_d \\ 0\end{bmatrix}\right) \\ &= \frac{1}{2}\left(\overline{\boldsymbol{\delta v}}^b \circ \overline{\boldsymbol{\delta q}} + \overline{\boldsymbol{d}} \circ \overline{\boldsymbol{\delta\omega}}\right) + \frac{1}{2}\begin{bmatrix}\Sigma(\overline{\boldsymbol{\delta p}}, \overline{\boldsymbol{\delta q}}) \\ 0\end{bmatrix}\boldsymbol{\omega}_d,\end{aligned}$$

where $\begin{bmatrix}\Sigma(\overline{\boldsymbol{\delta p}}, \overline{\boldsymbol{\delta q}}) \\ 0\end{bmatrix}\boldsymbol{\omega}_d = \overline{\boldsymbol{\delta p}}^b \circ \begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta q})\boldsymbol{\omega}_d \\ 0\end{bmatrix} + \begin{bmatrix}\boldsymbol{S}(\boldsymbol{\delta p}^b)\boldsymbol{\omega}_d \\ 0\end{bmatrix} \circ \overline{\boldsymbol{\delta q}}$, follows from (3). Applying equation (3) to $\dot{\overline{\boldsymbol{d}}} = \dot{\mathcal{D}}(\delta\boldsymbol{Q})$, and $\boldsymbol{d} = \frac{1}{2}\left(\boldsymbol{S}(\boldsymbol{\delta p})\boldsymbol{\delta q} + \boldsymbol{\delta p}\delta q_0\right)$, the proof is completed. □

III. Control algorithm

In [19], an algorithm based on dual quaternions was proposed for a leader-follower problem, where the dynamics involved showed a behavior similar to equation (6). The proposed algorithm was a proportional controller, which took into account the structure of the dual quaternion dynamics. The main drawback of that strategy was the steady-state tracking error, which was evidenced in the experimental results [19].

In order to overcome this limitation, a new control algorithm for the dual quaternion dynamics is here proposed. For this purpose, unit norm quaternion $\overline{\boldsymbol{\eta}} = (\boldsymbol{\eta}, \eta_0) \in \mathbb{H}$ and $\boldsymbol{\xi} \in \mathbb{R}^3$ are introduced,

$$\dot{\overline{\boldsymbol{\eta}}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{S}(\boldsymbol{\eta}) + \boldsymbol{I}\eta_0 \\ -\boldsymbol{\eta}^T\end{bmatrix}\left(-|\delta q_0|\boldsymbol{K}_{\omega,i}\boldsymbol{\delta q} + \mathrm{sgn}(\eta_0)\boldsymbol{K}_\eta\boldsymbol{\eta}\right), \tag{7}$$

$$\dot{\boldsymbol{\xi}} = -\boldsymbol{K}_{v,i}\boldsymbol{\delta p}^b + \boldsymbol{K}_\xi\boldsymbol{\xi}, \tag{8}$$

where $\boldsymbol{K}_{\omega,i}, \boldsymbol{K}_\eta, \boldsymbol{K}_{v,i}, \boldsymbol{K}_\xi \in \mathbb{R}^{3\times3}$ are negative definite and $\mathrm{sgn}(x)$ is the sign function such that $\mathrm{sgn}(0) = 1$. These additional terms compensate the tracking error, with a forgetting factor given by gain matrices $\boldsymbol{K}_\xi, \boldsymbol{K}_\eta$. The following theorem

states the angular velocity and linear velocity that must be applied to the vehicle in order to achieve the desired attitude and position, i.e., $(\boldsymbol{\delta q}, \boldsymbol{\delta p}) \to \mathbf{0}$.

Theorem 1. *Let $\boldsymbol{K}_{\omega,p}, \boldsymbol{K}_{v,p} \in \mathbb{R}^{3\times3}$ be negative definite matrices, $\boldsymbol{\xi} \in \mathbb{R}^3$ and $\overline{\boldsymbol{\eta}} \in \mathbb{H}$ with kinematics as in equations* (7) *and* (8)*. Given $\boldsymbol{\omega}_d, \boldsymbol{v}_d$ and the (desired) dual quaternion $\boldsymbol{Q}_d$ satisfying* (5)*, suppose that the dual quaternion $\boldsymbol{Q}$ is given by equation* (5) *with*

$$\overline{\boldsymbol{\omega}} = (\boldsymbol{\omega}, 0) = (\boldsymbol{\omega}_d + \operatorname{sgn}(\delta q_0)(\boldsymbol{K}_{\omega,p}\boldsymbol{\delta q} + \eta_0 \boldsymbol{K}_{\omega,i}\boldsymbol{\eta}), 0) \quad (9)$$

$$\overline{\boldsymbol{v}} = (\boldsymbol{v}, 0) = (\boldsymbol{v}_d + \boldsymbol{R}(\overline{\boldsymbol{q}}_d)\,(\boldsymbol{K}_{v,p}\boldsymbol{\delta p}^b + \boldsymbol{K}_{v,i}\boldsymbol{\xi}), 0). \quad (10)$$

Then $(\overline{\boldsymbol{\delta q}}, \overline{\boldsymbol{\delta p}}, \boldsymbol{\xi}, \overline{\boldsymbol{\eta}}) \to \overline{\mathbf{0}}$, with error given by $\delta\boldsymbol{Q} = \boldsymbol{Q}_d^* \circ \boldsymbol{Q}$.

Proof. Being $V = V(\overline{\boldsymbol{\delta q}}, \overline{\boldsymbol{d}}, \boldsymbol{\xi}, \overline{\boldsymbol{\eta}}) \geq 0$ the Lyapunov function given by $V = \frac{1}{2}\boldsymbol{\delta q}^T\boldsymbol{\delta q} + \frac{1}{2}(1-\delta q_0^2) + 2\boldsymbol{d}^T\boldsymbol{d} + 2d_0^2 + \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\eta} + (1-\eta_0^2) + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}$, observe that it satisfies $V = 0$ if and only if $\boldsymbol{\delta q} = \boldsymbol{d} = \boldsymbol{\xi} = \boldsymbol{\eta} = \mathbf{0}$ and $1 - \delta q_0 = d_0 = 1 - \eta_0 = 0$. Taking the time derivative, by Lemma 1 and equation (7) it follows that

$$\begin{aligned}\dot{V} = {} & \frac{1}{2}\boldsymbol{\delta q}^T\left((\boldsymbol{S}(\boldsymbol{\delta q}) + \boldsymbol{I}\delta q_0)\boldsymbol{\delta\omega} + 2\boldsymbol{S}(\boldsymbol{\delta q})\boldsymbol{\omega}_d\right) \\ & + \frac{1}{2}\boldsymbol{\delta q}^T\boldsymbol{\delta\omega}\delta q_0 + 2\boldsymbol{d}^T((\boldsymbol{S}(\boldsymbol{d}) + d_0\boldsymbol{I})\boldsymbol{\delta\omega} \\ & + (-\boldsymbol{S}(\boldsymbol{\delta q}) + \delta q_0\boldsymbol{I})\boldsymbol{\delta v}^b) + 4\boldsymbol{d}^T\boldsymbol{S}(\boldsymbol{d})\boldsymbol{\omega}_d \\ & - 2d_0(\boldsymbol{d}^T\boldsymbol{\delta\omega} + \boldsymbol{\delta q}^T\boldsymbol{\delta v}^b) + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\omega}_\xi \\ & + \frac{1}{2}\boldsymbol{\eta}^T(\boldsymbol{S}(\boldsymbol{\eta}) + \eta_0\boldsymbol{I})\boldsymbol{\omega}_\eta + \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{\omega}_\eta\eta_0,\end{aligned}$$

where $\boldsymbol{\omega}_\xi = -\boldsymbol{K}_{v,i}\boldsymbol{\delta p}^b + \boldsymbol{K}_\xi\boldsymbol{\xi}$ and $\boldsymbol{\omega}_\eta = -|\delta q_0|\boldsymbol{K}_{\omega,i}\boldsymbol{\delta q} + \operatorname{sgn}(\eta_0)\boldsymbol{K}_\eta\boldsymbol{\eta}$. Since $\boldsymbol{x}^T\boldsymbol{S}(\boldsymbol{x}) = \mathbf{0}$ for every $\boldsymbol{x} \in \mathbb{R}^3$ and canceling out some terms, it follows that $\dot{V} = \boldsymbol{\delta q}^T\boldsymbol{\delta\omega}\,\delta q_0 + 2(\boldsymbol{d}^T(-\boldsymbol{S}(\boldsymbol{\delta q}) + \delta q_0\boldsymbol{I}) - d_0\boldsymbol{\delta q}^T)\,\boldsymbol{\delta v}^b + \boldsymbol{\xi}^T\boldsymbol{\omega}_\xi + \eta_0\,\boldsymbol{\eta}^T\boldsymbol{\omega}_\eta$. Now, observe that $\overline{\boldsymbol{\delta p}}^b = 2\overline{\boldsymbol{d}} \circ \overline{\boldsymbol{\delta q}}^*$, by equation (3) $\boldsymbol{\delta p}^b = 2\,(\boldsymbol{S}(\boldsymbol{\delta q})\boldsymbol{d} + \boldsymbol{d}\delta q_0 - d_0\boldsymbol{\delta q})$ then, $\dot{V} = \boldsymbol{\delta q}^T\boldsymbol{\delta\omega}\,\delta q_0 + \boldsymbol{\delta p}^{b^T}\boldsymbol{\delta v}^b + \boldsymbol{\xi}^T(-\boldsymbol{K}_{v,i}\boldsymbol{\delta p}^b + \boldsymbol{K}_\xi\boldsymbol{\xi}) + \eta_0\,\boldsymbol{\eta}^T(-|\delta q_0|\boldsymbol{K}_{\omega,i}\boldsymbol{\delta q} + \operatorname{sgn}(\eta_0)\boldsymbol{K}_\eta\boldsymbol{\eta})$.

By definition of $\boldsymbol{\omega} = \boldsymbol{\omega}_d + \boldsymbol{\delta\omega}$, $\boldsymbol{v} = \boldsymbol{v}_d + \boldsymbol{R}(\overline{\boldsymbol{q}}_d)\boldsymbol{\delta v}^b$ and canceling out some terms, it follows that $\dot{V} = |\delta q_0|\boldsymbol{\delta q}^T\boldsymbol{K}_{\omega,p}\boldsymbol{\delta q} + \boldsymbol{\delta p}^{b^T}\boldsymbol{K}_{v,p}\boldsymbol{\delta p}^b + \boldsymbol{\xi}^T\boldsymbol{K}_\xi\boldsymbol{\xi} + |\eta_0|\boldsymbol{\eta}^T\boldsymbol{K}_\eta\boldsymbol{\eta} \leq 0$.

Notice that, if $\dot{V} = 0$, then $\boldsymbol{\delta p}^b = |\eta_0|\,\boldsymbol{\eta} = \boldsymbol{\xi} = |\delta q_0|\boldsymbol{\delta q} = 0$. However, if $\delta q_0 = \mathbf{0}$, by equation (6), it follows that $\mathbf{0} = -\boldsymbol{\delta q}^T\boldsymbol{K}_{\omega,p}\boldsymbol{\delta q}$, and this is not possible because $\boldsymbol{K}_{\omega,p} < 0$ and $\|\overline{\boldsymbol{\delta q}}\| = 1$, then $\boldsymbol{\delta q} = \mathbf{0}$. Similarly it can be shown that $\eta_0 = 0$ cannot satisfy $\dot{V} = 0$, then $(\boldsymbol{\delta q}, \boldsymbol{\delta p}, \boldsymbol{\xi}, \boldsymbol{\eta}) \to \mathbf{0}$. □

IV. Dual Quaternion Formation Definition

In this section, we adopt a cluster-space control-based methodology for the specification of the robot formation. A new space of cluster variables can be specified based on geometric characteristics of the formation. Then, a controller that operates on this new space —reducing the errors accordingly— is proposed. Compensation signals generated by the controller are then transformed to the space of the vehicles to be applied to the system. Figure 3 shows the architecture of the controller. To illustrate this, consider the following example where three robots are flying in a given plane with a prescribed configuration, as shown in Figure 2. Here, the idea is to specify the position of a leader $\boldsymbol{p_1}$ (robot 1), the angle φ with respect to its followers (robots 2 and 3), and the relative position between them. The orientation of the robot formation is given by the vector pointing from the center of mass of the formation (cm) to the leader robot. A desired orientation is given by the unit vector $\boldsymbol{m}$. For this particular application, instead of working in the robot space, it would be more appropriated to define the cluster variables in term of geometric characteristics of the formation. This can be done in terms of dual quaternions as follows. The flying plane $\boldsymbol{\Pi}$ can be defined with the normal unit vector $\boldsymbol{n}$ and the position of the leader robot $\boldsymbol{p_1}$, by defining the dual quaternion $\boldsymbol{Q}_{c_1} = \overline{\boldsymbol{q_{c_1}}} + \frac{\varepsilon}{2}(\overline{\boldsymbol{p_1}} \circ \overline{\boldsymbol{q_{c_1}}})$, where $\overline{\boldsymbol{q_{c_1}}} = (\boldsymbol{n}\sin(\phi/2), \cos(\phi/2))$. This dual quaternion gives geometric characteristics of the formation. We can compute this dual quaternion based on robots positions, in fact: $\overline{\boldsymbol{q_{c_1}}} = (\frac{\boldsymbol{S(r_1)r_3}}{\|\boldsymbol{r_2}+\boldsymbol{r_3}\|}, \frac{1+\boldsymbol{r_2}^T\boldsymbol{r_3}}{\|\boldsymbol{r_2}+\boldsymbol{r_3}\|})$, where $\boldsymbol{r_k} = \frac{\boldsymbol{p_k}-\boldsymbol{p_1}}{\boldsymbol{p_k}-\boldsymbol{p_1}}$, $k = 2, 3$. To complete the formation specification, we need the dual quaternion $\boldsymbol{Q}_{c_2} = \overline{\boldsymbol{q_{c_2}}} + \frac{\varepsilon}{2}((\overline{\boldsymbol{p_3}} - \overline{\boldsymbol{p_2}}) \circ \overline{\boldsymbol{q_{c_2}}})$, where $\overline{\boldsymbol{q_{c_2}}} = (\boldsymbol{n}\sin(\alpha/2), \cos(\alpha/2))$, and $\overline{\boldsymbol{q_{c_2}}} = (\frac{\boldsymbol{S(u)m}}{\|\boldsymbol{u}+\boldsymbol{m}\|}, \frac{1+\boldsymbol{u}^T\boldsymbol{m}}{\|\boldsymbol{u}+\boldsymbol{m}\|})$, where $\boldsymbol{u} = \boldsymbol{p_1} - \boldsymbol{p_{cm}}$. Notice that $\boldsymbol{Q}_{c_1}$ contains information of the flying plane and relative angle between leader and followers, while $\boldsymbol{Q}_{c_2}$ captures the formation orientation and relative position between follower robots.

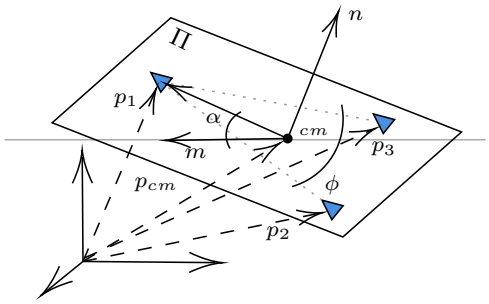


Fig. 2: Dual quaternion-based cluster definition.

As an alternative definition of a robot cluster, the leader-follower strategy can be included within the same formulation, where the space is composed of the pose (position and attitude) of the leader and the relative pose of the followers with respect to the leader.

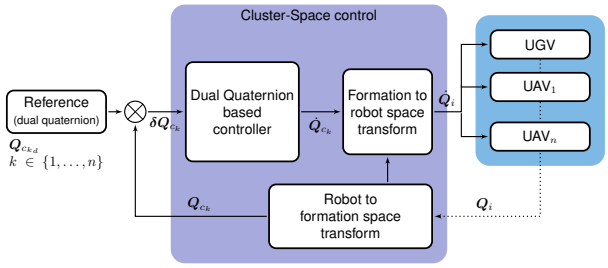


Fig. 3: Cluster-space dual quaternion controller block diagram.

From this perspective, and given a dual quaternion representation of the i^{th} robot's position and orientation: $\boldsymbol{Q}_i = \overline{\boldsymbol{q}}_i + \varepsilon\frac{1}{2}\overline{\boldsymbol{p}}_i \circ \overline{\boldsymbol{q}}_i$ where $\overline{\boldsymbol{q}}_i \in \mathbb{H}$ is the quaternion representing the orientation and $\overline{\boldsymbol{p}}_i = (x_i, y_i, z_i, 0)$ represents its position.

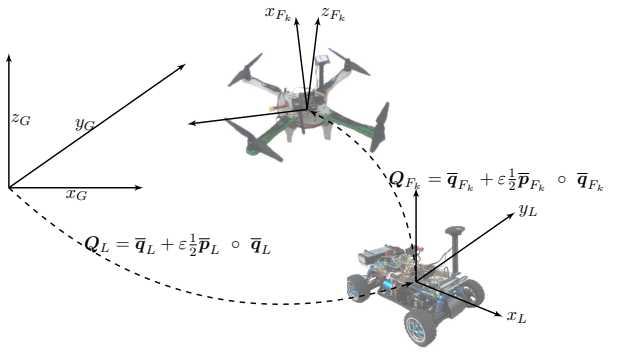


Fig. 4: Proposed leader-follower formation definition based on dual quaternions.

Without loss of generality, designating robot 1 as a leader and robot $k = 2, \ldots, N$ as a follower, it is possible to define the relative leader-follower pose,

$$\boldsymbol{Q}_L = \boldsymbol{Q}_1, \;\; \boldsymbol{Q}_{F_k} = \boldsymbol{Q}_k \circ \boldsymbol{Q}_1^* \text{ for every } k = 2, \ldots, N. \quad (11)$$

Equations (11) can be seen as a kinematic transformation

$$\mathbf{FWD}(\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N) = \begin{bmatrix} \boldsymbol{Q}_L \\ \boldsymbol{Q}_{F_2} \\ \vdots \\ \boldsymbol{Q}_{F_N} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_N) \\ f_2(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_N) \\ \vdots \\ f_N(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_N) \end{bmatrix} \quad (12)$$

that relates the representation in the space of the robots to the ones in the space of the formation. Here $\mathbf{FWD}$ denotes the forward transformation (1) and $\boldsymbol{r} = (\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N)$. Figure 4 show these relations. The relation between the velocities is given by the Jacobian matrix $\boldsymbol{J}(\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N)$ associated to $\mathbf{FWD}$. Additionally, an inverse kinematic relation is defined as $\boldsymbol{Q}_1 = \boldsymbol{Q}_L, \;\; \boldsymbol{Q}_k = \boldsymbol{Q}_{F_k} \circ \boldsymbol{Q}_L$, for every $k = 2, \ldots, N$.

One advantage of working with the space of the formation is that if the formation or the task change, the same control architecture can be used, modifying the state definition accordingly. More specifically, in [18] the formation was defined as an alternative robot cluster instead of a leader-follower scheme. That is, in [18], a formation of two robots is represented as a segment that can be rotated, moved and scaled over time, which can be considered as an alternative definition to the leader-follower specification when the task at hand makes it more convenient. The results presented here can therefore be extended to any robot cluster definition, such as the two presented above. The main reason to develop common schemes for different robot formation definitions is that it simplifies the development of the software needed for deployment and facilitates switching between different strategies. Of course, for each cluster definition, the function $\mathbf{FWD}(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_N)$ must be defined, as well as the associated Jacobian matrix $\dot{\mathbf{FWD}} = \boldsymbol{J}(\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_N)[\dot{\boldsymbol{Q}}_1, \dot{\boldsymbol{Q}}_2, \ldots, \dot{\boldsymbol{Q}}_N]^T$. These expressions are used in the dual quaternion formation block diagram shown in Figure 3.

In [18] a control algorithm was developed for a cluster-space architecture, later in [19] the same idea was proposed for a leader-follower strategy, both based on dual quaternions. Next, we derive the leader-follower formation controller on dual quaternions, based on the previous cluster-space definition for compensation of the steady state tracking error observed in [19]. In order to do that, a new term compensating steady-state tracking errors that includes an integral term with a forgetting factor is introduced. The idea follows Theorem 1. The dual quaternion error is defined as

$$\delta\boldsymbol{Q}_m = \boldsymbol{Q}_m^* \circ \boldsymbol{Q}_{m_d}; \quad m = \{L, F_k\}, \quad (13)$$

where $\boldsymbol{Q}_{m\,d}$ are the desired leader position and orientation and followers relative position and orientations. Theorem 2 gives stability guarantee on the pose error of the leader vehicle, and the relative position of the followers with respect to the leader.

Theorem 2. *For each $m = \{L, F_k\}, k = 2, \ldots, N$, assume that the desired dual quaternion $\boldsymbol{Q}_{m_d}$, angular velocity $\boldsymbol{\omega}_{m_d}$ and linear velocity $\boldsymbol{v}_{m_d}$ are given. Suppose that $\boldsymbol{K}_{\omega,p_m}$, $\boldsymbol{K}_{\omega,i_m}$, $\boldsymbol{K}_{v,p_m}$, $\boldsymbol{K}_{v,i_m} \in \mathbb{R}^{3\times3}$ are strictly negative definite matrices, and let $\delta\boldsymbol{Q}_m = (\delta\boldsymbol{q}_m, \delta q_{0_m}, \delta\boldsymbol{d}_m, \delta d_{0_m})$ be the tracking error defined in equation* (13)*. If the control commands ($\boldsymbol{Q}_{mc}$) defined for the leader and the followers are*

$$\dot{\boldsymbol{Q}}_{mc} = \frac{1}{2}\begin{bmatrix} \boldsymbol{S}(\boldsymbol{q}_{mc}) + \boldsymbol{I}q_{0\,mc} & 0 \\ -\boldsymbol{q}_{mc}^T & 0 \\ \boldsymbol{S}(\boldsymbol{d}_{mc}) + \boldsymbol{I}d_{0\,mc} & -\boldsymbol{S}(\boldsymbol{q}_{mc}) + \boldsymbol{I}q_{0\,mc} \\ -\boldsymbol{d}_{mc}^T & -\boldsymbol{q}_{mc}^T \end{bmatrix}\begin{bmatrix} \boldsymbol{\omega}_{mc} \\ \boldsymbol{v}_{mc}^b \end{bmatrix}$$

$$\boldsymbol{\omega}_{mc} = \boldsymbol{\omega}_{m_d} + \mathrm{sgn}(\delta q_{0_m})(\boldsymbol{K}_{\omega,p_m}\delta\boldsymbol{q}_m + \eta_{0_m}\boldsymbol{K}_{\omega,i_m}\boldsymbol{\eta}_m),$$
$$\boldsymbol{v}_{mc} = \boldsymbol{v}_{m_d} + \boldsymbol{R}(\overline{\boldsymbol{q}}_{m_d})\,(\boldsymbol{K}_{v,p_m}\boldsymbol{R}(\overline{\boldsymbol{q}}_{m_d}^*)\delta\boldsymbol{p}_m + \boldsymbol{K}_{v,i_m}\boldsymbol{\xi}_m).$$

$$\dot{\overline{\boldsymbol{\eta}}}_m = \frac{1}{2}\begin{bmatrix} \boldsymbol{S}(\boldsymbol{\eta}_m) + \boldsymbol{I}\eta_{0_m} \\ -\boldsymbol{\eta}_m^T \end{bmatrix}(-|\delta q_{0_m}|\boldsymbol{K}_{\omega,i_m}\delta\boldsymbol{q}_m + \mathrm{sgn}(\eta_{0_m})\boldsymbol{K}_{\eta_m}\boldsymbol{\eta}_m), \quad (14)$$

$$\dot{\overline{\boldsymbol{\xi}}}_m = -\boldsymbol{K}_{v,i_m}\boldsymbol{R}(\overline{\boldsymbol{q}}_{m_d}^*)\delta\boldsymbol{p}_m + \boldsymbol{K}_{\xi_m}\boldsymbol{\xi}_m \quad (15)$$

then, $\lim_{t\to\infty}(\delta\mathbf{q}_m, \delta\mathbf{d}_m) = (0, 0)$, for every $m = \{L, F_k\}$.

The proof is omitted for space limitations. It follows the same reasoning as Theorem 1 but using the Lyapunov function

$$V = \sum_{m=\{L,F_k\}} \frac{1}{2}\delta\boldsymbol{q}_m^T\delta\boldsymbol{q}_m + \frac{1}{2}(1 - \delta q_{0_m}^2) + 2\delta\boldsymbol{d}_m^T\delta\boldsymbol{d}_m + 2d_{0_m}^2 + \frac{1}{2}\boldsymbol{\eta}_m^T\boldsymbol{\eta}_m + (1 - \eta_{0_m}^2) + \frac{1}{2}\boldsymbol{\xi}_m^T\boldsymbol{\xi}_m.$$

V. Experimental Results

In this section, results from numerical simulations and real experimental data are shown. First, the proposed algorithm is implemented and executed under the same conditions and with the same trajectories on both the simulation and the experimental setup. This not only allows for the verification of the algorithms and the simulator, but it also enables its use to predict the behaviour of the real system. Then, results with the proposed controller are presented.

In [19], the need of an algorithm capable of compensating the steady-state error was shown. Using numerical simulations, it is possible to see how the controller proposed in this work improves the tracking performance. To test the advantage of the proposed controller with the same conditions and disturbances, we first run numerical simulations using ROS and Gazebo. The UAV model was the IRIS from PX4 gazebo SITL. For the UGV model, the MIT RACECAR plugin for Gazebo was

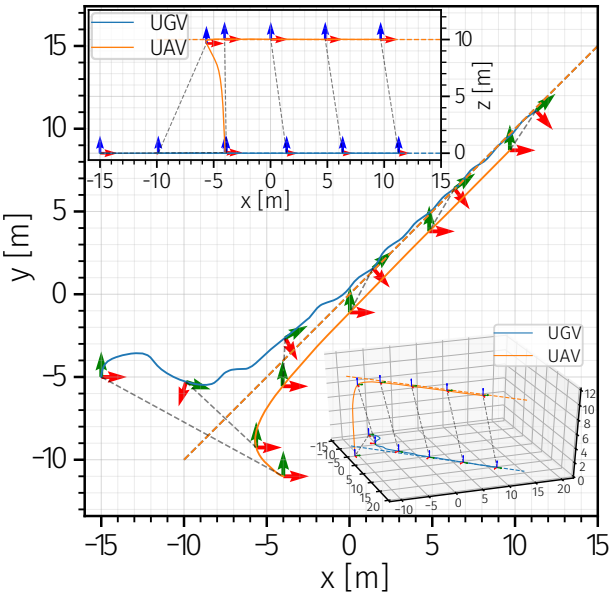


Fig. 5: Simulation results of a UGV following a rectilinear trajectory and a UAV flying above it at a constant relative altitude of $10\,\mathrm{m}$. **Main figure:** XY plane top view. **Insets:** Isometric view (bottom), XZ plane side view (top).

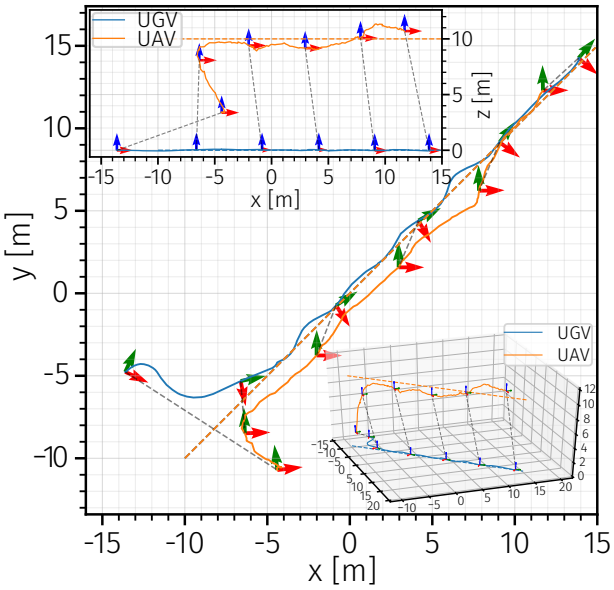


Fig. 6: Experimental result from [19] of a UGV following a rectilinear trajectory and a UAV flying above it at a constant relative altitude of $10\,\mathrm{m}$. Comparable with Fig. 5. **Main figure:** XY plane top view. **Insets:** Isometric view (bottom), XZ plane side view (top).

used. The host operating system was Ubuntu, running on a laptop computer. A port of the PX4 autopilot firmware was compiled on host and the communication with the IRIS model was done with a mavros interface.

To validate the numerical simulator, a ROS/Gazebo simulation recreates experimental results previously reported in [19], with the same references and controller used with the real vehicles. In this first experiment, a quadcopter UAV and a 1/16th scale RC buggy chassis UGV were used. Both vehicles had a Pixhawk Flight Controller with autopilot firmware PX4 v1.6.5. Both vehicles were connected to the host and interfaced using mavros. The new controller proposed in Theorem 2 was implemented using the computational library DQ Robotics [21] for dual quaternion algebra and connected to the ROS environment [22].

A. Validation of the Simulation Environment

The first simulation consists of the UGV (formation leader) following a rectilinear trajectory while the UAV flies over it, keeping the relative position constant and a fixed orientation in the global frame. The UAV flies at a distance of $10\,\mathrm{m}$ and an elevation angle of $90°$, meaning 10 meters right over the UGV. In this simulation, a dual quaternion-based proportional controller is used to control all the vehicles in the formation. Control gains are set to $\boldsymbol{K}_{w,p_m} = 0.3\boldsymbol{I}_3$ and $\boldsymbol{K}_{v,p_m} = 0.3\boldsymbol{I}_3$ and $\boldsymbol{K}_{\omega,i} = \boldsymbol{K}_{v,i} = 0$, in order to consider only the proportional effect. Figure 5 shows the vehicles' motion through the test, from different perspectives.

It can be seen that both vehicles track their references. It can also be noted a lag between the position of the UGV and that of the UAV. This lag is produced by the steady-state error, as expected for a pure proportional controller.

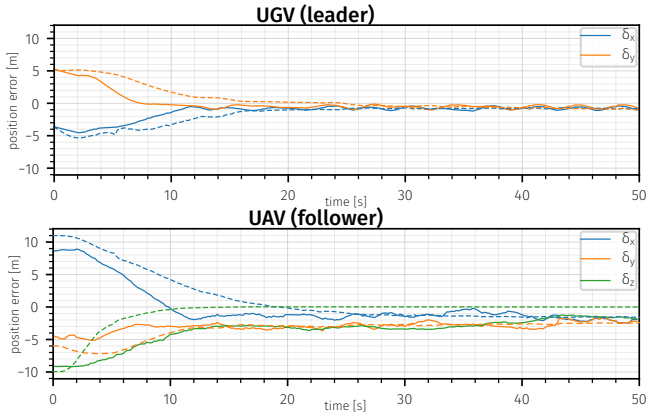
In the corresponding field experiment, reproduced here in Figure 6 for direct comparison, the controller gains used were the same as those used in simulation. Figure 5 shows the path followed by the UGV and the UAV. Figures 7a and 7b show the vehicles' position errors and the UAV yaw error, for the field experiment and the simulation. Yaw error for the UGV is not estimated, as its orientation is computed from the commanded motion, due to nonholonomic constraints, therefore, there's no desired yaw angle.

One of the advantages of having a good numerical simulator is that we can evaluate the performance of our algorithms in a controlled environment. Additionally, this numerical simulator can be used to adjust controller gains prior to field experiments.

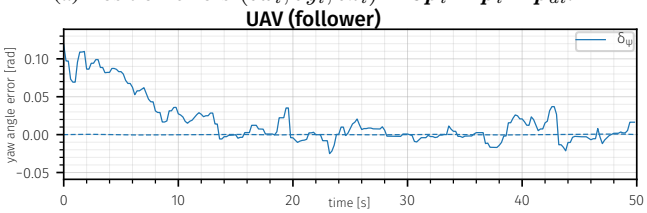
B. Steady-state error mitigation

To validate the effectiveness of the algorithm proposed in this work, a numerical simulation is presented where the integral part of the controller is activated at a certain point in time to highlight its impact on the control system performance. Performing a simulation of this scenario allows to maintain disturbances constants throughout the experiment. Figure 8a shows the position and orientation errors for a simulation result in the ROS/Gazebo environment where the additional term improves the performance of the controller presented in [19].

In the simulation, a formation of a leader UGV and two follower UAV performs a position and orientation regulation, where the vehicles track a constant reference. From $t = 0\,\mathrm{s}$ to $t = 1000\,\mathrm{s}$ only the proportional term of the controller is active. A bias in the position error of the two UAV is evident during this period. At $t = 1000\,\mathrm{s}$ the integral term of the controller is also activated, effectively reducing the error bias as expected.



(a) Position errors $(\delta x_i, \delta y_i, \delta z_i) = \boldsymbol{\delta p}_i = \boldsymbol{p}_i - \boldsymbol{p}_{di}$.



(b) Orientation (yaw) error $(\delta\psi_i)$.

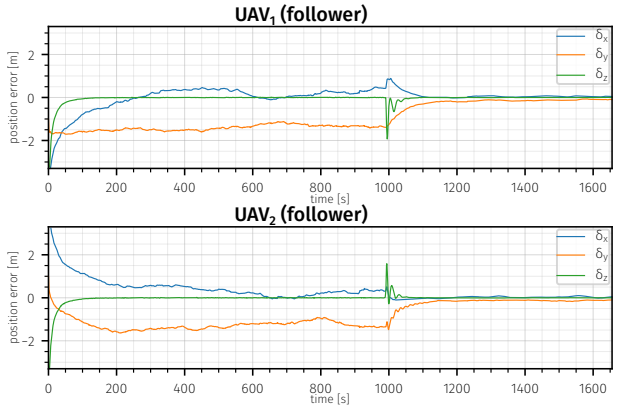
Fig. 7: Simulation (dashed) and experimental (solid) results of a UGV following a rectilinear trajectory and a UAV flying above it at a constant relative altitude of $10\,\mathrm{m}$.

Position errors benefit significantly from the addition of the integral term. It should also be noted that although a non-negligible transient is seen when the control algorithm is switched, this phenomenon is an artifact that would not be seen in normal operation, as the integral part of the controller would be active at all times. It is also worth noting that the high frequency components of the signals decrease as it would be expected from the low-pass filter nature of integral control.

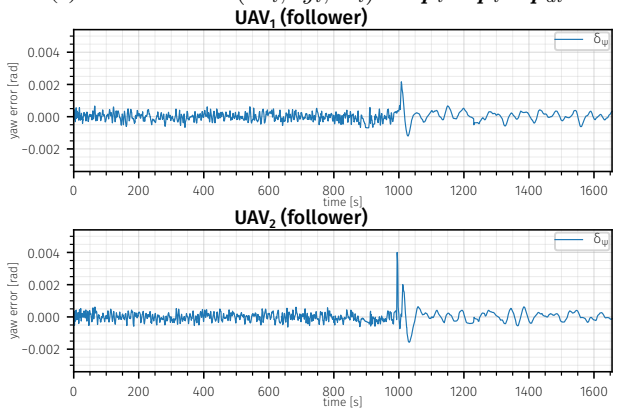
C. Experimental validation

In order to validate the control algorithm in a more interesting scenario, we performed experiments with two UAV in a leader-follower configuration, as Figure 1 shows. The reference trajectory for the leader robot is an 8-shape (lemniscate of Gerono). The follower robot is trying to maintain a constant relative position with respect to leader, as seen in Figure 9. The parameters of the controller were established based on simulations, resulting the following values: $K_{v,p} = -1.3\boldsymbol{I}_3$, $K_{v,i} = -1.2\boldsymbol{I}_3$, $K_\xi = -10^{-3}\boldsymbol{I}_3$, $K_{\omega,p} = -0.6\boldsymbol{I}_3$, $K_{\omega,i} = -0.2\boldsymbol{I}_3$, $K_\eta = -10^{-3}\boldsymbol{I}_3$.

Figure 10 shows the error between the leader robot and the commanded trajectory, and also the relative position error between the follower and the leader. Over green background is the time when the closed-loop control is activated, and over blue background when the vehicles are manually controlled by the pilot, during take-off and landing. Figure 11 shows the yaw angle tracking error for both UAVs. When the control algorithm is activate, it takes about 5 seconds for the follower vehicle to follow the reference. After this transient, both vehicles follow the reference with an error below $1.5°$ during the experiment.



(a) Position errors $(\delta x_i, \delta y_i, \delta z_i) = \boldsymbol{\delta p}_i = \boldsymbol{p}_i - \boldsymbol{p}_{di}$.



(b) Orientation (yaw) error $(\delta\psi_i)$.

Fig. 8: Simulation results of a formation of one UGV and two UAV regulating their positions with the proposed controller. At $t = 1000\,\mathrm{s}$ the control algorithm is switched.

VI. Final Discussion

Formation control of multi-robot systems is an area of research with potential for several applications. Different control schemes, such as cluster control, are proposed as a method to simplify the control vehicles. One of the most used strategies for formation control of vehicles is the leader-follower approach, which facilitates the control of the formation commanding relative vehicle poses. In this work we have shown that it is possible to establish a common scheme that treats the leader-follower strategy and cluster control strategies within the same framework. This may be attractive when developing software for formation flying control that can switch from one strategy to another in mission time. This is why the leader-follower strategy was presented as a kinematic transformation with its associated Jacobian matrices.

On the other hand, the dual quaternions are a useful tool for implementing control software —in spite of the complexity in extracting physical information from these variables— since, from the implementation point of view, these are very suitable, because the computations are cheaper than for rotation matrices, and also have a simpler type invariant structure making it easier to detect numerical errors.

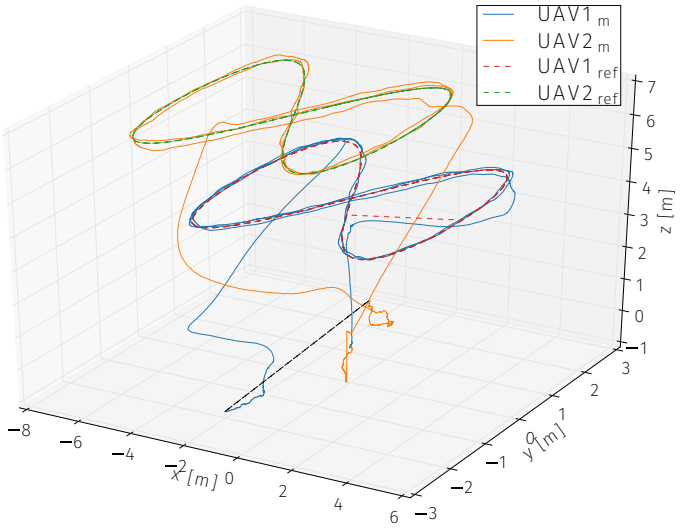


Fig. 9: Trajectories of two UAVs flying in formation: references and measured trajectories.

Fig. 10: Position error. UAV1 with respect to the reference trajectory (Top) and UAV2 with respect to the leader vehicle UAV1 (Bottom).

Notice that, with the control algorithm proposed in this paper, we can achieve a good performance by reducing the steady-state tracking error that was noticed in the experimental results in [19]. In terms of system architecture, the choice of the hardware/software suite used to develop the testbed as well as the numerical simulator permits to validate the algorithms in a very simple way, allowing to move from the computational simulations to a real application without requiring important modifications.

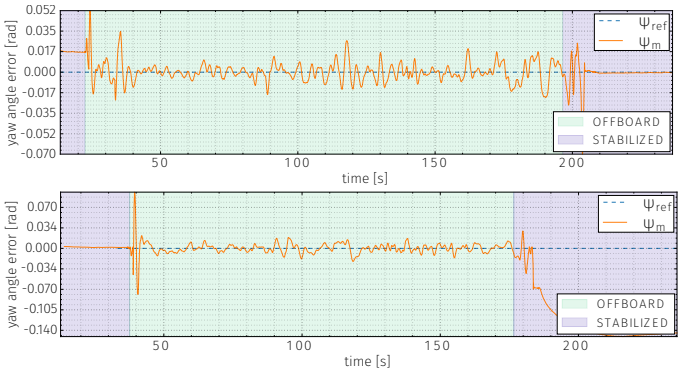


Fig. 11: Attitude error: Yaw angle error of UAV1 (Top) and UAV2 (Bottom).

REFERENCES

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed control of robotic networks. A mathematical approach to robot coordination algorithms*. Princeton University Press, 2009.

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

[3] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent network*. Princeton University Press, 2010, vol. 33.

[4] C. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 14(2), pp. 207–218, 2009.

[5] G. Michieletto, A. Cenedese, and A. Franchi, "Bearing rigidity theory in se (3)," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5950–5955.

[6] F. Schiano and R. Tron, "The dynamic bearing observability matrix nonlinear observability and estimation for multi-agent systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3669–3676.

[7] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Robust formation control in se (3) for tree-graph structures with prescribed transient and steady state performance," *Automatica*, vol. 103, pp. 538–548, 2019.

[8] Y. Wang and C. Yu, "Distributed attitude and translation consensus for networked rigid bodies based on unit dual quaternion," *Int. J. Robust and Nonlinear Control*, 2017.

[9] P. van Goor, Z. Sun, and C. Yu, "Attitude and pose formation control for multiple 3d rigid bodies based on unit quaternion representation," in *2018 37th Chinese Control Conference*. IEEE, 2018, pp. 6505–6510.

[10] X. Wang and C. Yu, "Unit dual quaternion-based feedback linearization tracking problem for attitude and position dynamics," *Systems & Control Letters*, vol. 62, no. 3, pp. 225–233, 2013.

[11] H. J. Savino, L. C. Pimenta, J. A. Shah, and B. V. Adorno, "Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators," *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 142–178, 2020.

[12] J. Dooley and J. McCarthy, "On the geometric analysis of optimum trajectories for cooperating robots using dual quaternion coordinates," in *IEEE Intl. Conf. Robotics and Automation*, vol. 1, pp. 1031–1036.

[13] B. V. Adorno and P. Fraisse, "The cross-motion invariant group and its application to kinematics," *IMA Journal of Mathematical Control and Information*, vol. 34, no. 4, pp. 1359–1378, 2017.

[14] K. Lynch and F. Park, *Modern Robotics*. Cambridge Univ. Press., 2017.

[15] B. P. Malladi, E. Butcher, and R. G. Sanfelice, "Rigid body pose hybrid control using dual quaternions: Global asymptotic stabilization and robustness," *Journal of Guidance, Control, and Dynamics*, 2020.

[16] H. T. Kussaba, L. F. Figueredo, J. Y. Ishihara, and B. V. Adorno, "Hybrid kinematic control for rigid body pose stabilization using dual quaternions," *Journal of the Franklin Institute*, vol. 354, no. 7, pp. 2769–2787, 2017.

[17] B. V. Adorno, "Robot kinematic modeling and control based on dual quaternion algebra—part i: Fundamentals." 2017.

[18] I. Mas and C. Kitts, "Quaternions and dual quaternions: Singularity-free multirobot formation control," *J. Intell. & Robot. Sys.*, pp. 1–18, 2016.

[19] I. Mas, P. Moreno, J. Giribet, and D. V. Barzi, "Formation control for multi-domain autonomous vehicles based on dual quaternions," in *IEEE Int. Conf. Unmanned Aircraft Systems*, 2017, pp. 723–730.

[20] I. Mas and C. Kitts, "Dynamic control of mobile multirobot systems: the cluster space formulation," *IEEE Access*, vol.2, no.2, pp.558–570, 2014.

[21] B. V. Adorno and M. Marques Marinho, "DQ robotics: A library for robot modeling and control," *IEEE Robot. Autom. Mag.*, 2020.

[22] [Online]. Available: https://youtu.be/9Ti0cFH_0YY