Hierarchical Task-Based Formation Control and Collision Avoidance of UAVs in Finite Time

Miguel A. Trujillo[a], Héctor M. Becerra[b,*], David Gómez-Gutiérrez[c,d], Javier Ruiz-León[a], Antonio Ramírez-Treviño[a]

[a] *CINVESTAV Unidad Guadalajara, Av. del Bosque 1145 , 45019, Zapopan, Jalisco, México*
[b] *Centro de Investigación en Matemáticas (CIMAT), A.C., Jalisco S/N, Col. Valenciana, 36023, Guanajuato, Gto., México*
[c] *Multi-agent Autonomous Systems Lab, Intel Labs, Intel Tecnología de México, Av. del Bosque 1001, 45019, Zapopan, Jalisco, México*
[d] *Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Av. General Ramón Corona 2514, 45201, Zapopan, Jalisco, México*

Abstract

This work addresses the problem of distributedly controlling the position of a group of Unmanned Aerial Vehicles (UAVs), that are considered as holonomic three-dimensional agents, to achieve a desired formation while collisions are avoided. We propose a generic control scheme computed by the adaptive convex combination of two control laws dealing with two, possibly conflicting, tasks, which are formation control and obstacle avoidance. The main contributions of the paper are threefold: First, the whole proposed scheme is distributed by nature, since the formation control is formulated as a consensus problem of virtual agents and the collision avoidance strategy is reactive, valid for unknown environments. Second, two control protocols are proposed to guarantee convergence to the desired formation in finite or predefined time; in the last case, the formation is achieved in a constant time independently of the initial UAVs positions. And finally, the control scheme generates continuous control signals in spite of activation and deactivation of the obstacle avoidance task, and stability is proved even in the transition of tasks. The effectiveness of the proposed approach is shown through realistic simulations and real experiments

*Corresponding author
Email address: hector.becerra@cimat.mx (Héctor M. Becerra)

for a group of quadrotors.

Keywords: Formation control, consensus, multi-agent system, task-based control, unmanned aerial vehicles (UAVs)

1. Introduction

The coordination and control of Multi-Agent Systems (MAS) have attracted the attention of the research community in the last years (see for instance [1], [2], [3], [4]), due to the potential of a MAS to face complex tasks that a single agent is not able to handle. Distributed control approaches applied to a MAS require a communication network allowing to share information with a subset of agents (neighbors). In this context, several interesting problems and applications have been addressed in the literature, for instance, synchronization of complex networks ([5]), distributed resource allocation ([6]), consensus ([7]) and formation control of multiple agents ([8]). In this paper, we address the distributed formation control problem to drive a MAS of aerial vehicles to achieve a desired formation under time constraints, while avoiding collisions with each other and with static obstacles in the environment.

In the literature, one way to address the distributed formation control of a MAS has been as a consensus problem ([8]), where the prescribed relative positions are reached by exchanging only local information ([7, 9]). Linear consensus protocols with asymptotic convergence to a common agreement state were proposed in [7] and [9]. The authors demonstrated that the second smallest eigenvalue of the graph Laplacian (algebraic connectivity) determines the convergence rate of the MAS. Thus, the convergence to the agreement state in linear protocols is exponential and the settling time cannot be preset to fulfill a time constraint if required by an application.

In order to deal with time constraints in consensus applications, finite-time consensus protocols have been proposed, for instance in [10]. Nevertheless, the convergence time is an unbounded function of the initial disagreement among the agents. In fixed-time consensus protocols, the convergence time is uni-

formly bounded by a constant independently of the initial conditions; for example [11, 12, 13, 14]. However, such upper bound is often overestimated or even unknown. Thus, the design of protocols satisfying time constraints is challenging based on such results. To address these problems, consensus protocols with predefined-time convergence have been recently proposed in [15, 16, 17] for first-order systems and in [18] for second-order systems. Predefined-time convergence means that the settling time can be preset as required by the application and up to our knowledge, it has not been used before for formation control.

During the formation control, the agents may collide with each other and also with obstacles of the environment. Thus, a collision avoidance strategy must be used to guarantee the attainment of the formation task ([19]). The potential fields approach for collision avoidance has been exploited in MAS formation control, for instance in [20] and [21]. In the first work, a centralized algorithm based on potential fields is proposed for the obstacle avoidance task. In that approach the environment must be known. The second work presented a distributed scheme for formation control and collision avoidance for a team of wheeled leaders and followers, with the agents constrained to move following arcs of circle. A scheme for collision-free formation for a MAS of nonholomic robots without a leader and achieving consensus in orientation was proposed in [22]. In both last works, the collision avoidance strategy changes abruptly the trajectory of the agents and introduces discontinuities in their velocities. A variety of strategies have been proposed to address the formation control with obstacle avoidance for aerial vehicles. A method based on the artificial potential field approach and binary maps for Unmanned Aerial Vehicles (UAVs) modeled in 2D is proposed in [23]; the algorithm is centralized and requires parallel computation. A fuzzy logic behavior-based approach where each UAV evaluates the surrounding points to select the direction for step motion is proposed in [24]; broadcast of positions and velocities to use global coordinates is required. In [25] obstacle avoidance and formation reconfiguration is tackled using interfered fluid dynamical systems that consider the kinematic model and constraints of the

agents in known environments; receding horizon control is used to adjust some control parameters. The previous works for UAVs do not present a stability analysis of the control system. A nonlinear controller based on the potential field method is proposed in [26] to simultaneously perform trajectory tracking, formation keeping and collision avoidance; stability is proved but it has the limitation of requiring a broadcast transmission of position data.

In the robotics community, the task-function approach has been widely used to address control problems involving several contradictory objectives, captured by hierarchically organized tasks ([27]). This approach is used in [28, 29] as a centralized strategy to achieve robot formation controlling two tasks: the average of the agent's state and the variance of the states into desired values. These works were extended in [30] and [31] to include an obstacle avoidance task for 3D autonomous vehicles formation. A scheme based on null space of hierarchical tasks, where the tasks are control of a triangular formation using a centralized geometric method and collision avoidance, is presented in [30]. The scheme presented in [31] is based on null space behavioral with consensus for the formation and potential fields for the collision avoidance. Nevertheless, the first work is only applicable for groups of three agents while in the second work, based on potential fields, it has the requirement that the environment must be known. Furthermore, as in most of the previously mentioned works, the computed control inputs present discontinuities due to the switching schemes between the formation control and the evasion action.

In this paper, we tackle the two-tasks problem of *distributed* formation control with collision avoidance in the framework of hierarchical task-based control, which allows a theoretical stability analysis. The formation control is addressed as a consensus problem of virtual agents, related to the position of the real agents by a displacement vector, such that if the virtual agents reach consensus, then the real agents reach a desired formation in a distributed fashion. The collision avoidance becomes the task with higher priority when an obstacle gets closer than a given security distance, which can be other agent or a fixed obstacle in the environment. The collision avoidance control only requires local informa-

tion and thus, the proposed method is valid for unknown environments. In this work, the agents of the MAS are UAVs modeled as holonomic systems moving in 3D, specifically quadrotors, and for that, we have considered an ellipsoidal safety region to avoid disturbances caused by the turbulence of other agents. We introduce in the hierarchical task-based scheme that the UAVs achieve formation in a desired constant time independently of the initial conditions, based on predefined-time consensus ([16]). Besides the guarantee of constrained convergence time and the distributed nature of the proposed approach, the main difference and challenge with respect to existing protocols is that the switching between two control laws is treated by using a smooth transition, in such a way that the proposed control scheme is a convex combination of two control laws that generates continuous control inputs. Moreover, it is not needed to know a map of the environment and the obstacle avoidance strategy can be applied for irregular obstacles.

We have formulated an adequate task function for the formation control problem and it has been combined with the obstacle avoidance task in a hierarchical distributed control approach with its corresponding stability analysis. Thus, the main contributions of this work with respect to the existing literature are the following: a) The whole scheme is distributed by nature; the computation of the UAVs velocities depends only on information from its neighbors and the obstacle avoidance strategy is reactive, valid for unknown environments with irregular obstacles. b) Two alternative control protocols are proposed and integrated in the hierarchical task-based scheme such that formation is guaranteed to be achieved in finite or predefined time. In the finite-time case the formation is achieved in a bounded time that depends on the control gains used and the initial conditions, whereas in the predefined-time case the formation is achieved in a constant time independently of those factors. The predefined-time control law presents a convenient behavior, in particular for UAVs, since the generated control inputs have null initial value, evolve in a smooth way and return to zero at the convergence time. This property of the control law yields that the energy consumption in a real application is less than with existing fixed-time protocols

due to the lower control effort to achieve the desired formation. c) The proposed scheme provides continuity of the control inputs when both tasks are activated or deactivated, which is important for velocity control of quadrotors that requires smooth signals. To illustrate the effectiveness of the proposed approach, simulations are presented using a dynamic simulator and real experiments are conducted with a group of quadrotors.

This work is structured as follows. Section 2 recalls some definitions and results from graph theory. The task-based control scheme is also introduced in this section. In Section 3, the problem of formation control for UAVs with obstacle avoidance is formulated and solved using two different formation control laws. Section 4 evaluates the performance of the proposed approach in simulation and Section 5 presents results of real experiments with a group of quadrotors. Finally, the main conclusions of this work are presented in Section 6.

2. Preliminaries

2.1. Graph and Consensus Theory

In this section, some notations and preliminaries about graph and consensus theory are presented, an interested reader can consult literature as [32] and [7] for a deeper insight in the field. This paper focuses only on undirected graphs. In a network of agents, *consensus* means to reach an agreement on a certain quantity of interest that depends on the state of all agents ([7]). A model of a network is typically a graph $\mathcal{G}$, which consists of a vertex set $\mathcal{V}(\mathcal{G})$ and an edge set $\mathcal{E}(\mathcal{G})$. An edge is denoted by ij and $j \sim i$ denotes that the vertex i and vertex j are neighbors, i.e., there exists an edge ij. The set $\mathcal{N}_i(\mathcal{G}) = \{j : ji \in \mathcal{E}(\mathcal{G})\}$ represents the neighbors of vertex i in the graph $\mathcal{G}$. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ of a graph with N vertices is a square matrix with entries a_{ij} corresponding to the weight of the edge ij; when i is not adjacent to j then $a_{ij} = 0$. Through this work it is assumed that $a_{ij} = a_{ji}$, i.e. only *undirected* and *balanced graphs* are considered. The Laplacian matrix of $\mathcal{G}$ is $L(\mathcal{G}) = \Delta - A$

where $\Delta = diag(d_1, \cdots, d_N)$ with $d_i = \sum_{j=1}^{N} a_{ij}$. A sequence of distinct vertices starting with i and ending with j such that consecutive vertices are adjacent is called a *path* from i to j. A graph $\mathcal{G}$ is *connected* if there is a path between any two vertices, otherwise it is disconnected. If the graph $\mathcal{G}$ is connected, then the eigenvalue $\lambda_1(L) = 0$ has algebraic multiplicity one associated to the eigenvector $\mathbf{1} = [1 \ \cdots \ 1]^T$, i.e., $\ker L(\mathcal{G}) = \{\chi : \chi_1 = \ldots = \chi_N\}$. For undirected graphs the Laplacian matrix L is positive semidefinite and symmetric.

In this work, we will consider a MAS composed of N agents with decoupled single-integrator dynamics, connected with each other through a network, the agent's dynamics is given by

$$\dot{\chi}_i(t) = u_i, \ i \in \{1, \ldots, N\}, \tag{1}$$

where $\chi_i, u_i(t) \in \mathbb{R}^n$ are the state and the control input of agent i, respectively. In particular, we will consider a network of UAVs, where the decoupled three dimensional position of each UAV is the state χ_i and its translational velocities are the control input vector u_i ([33]). The network dynamics for agents (1) can be written in vector form as

$$\dot{\chi}(t) = u(t), \tag{2}$$

where $\chi(t) = [\chi_1(t), \ldots, \chi_N(t)]^T \in \mathbb{R}^{nN}$ is the state vector and $u(t) = [u_1(t), \ldots, u_N(t)]^T \in \mathbb{R}^{nN}$ is the control inputs vector of the network. The consensus error of agent i with respect to its neighbors is defined as ([7])

$$e_{c_i}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(\chi_j(t) - \chi_i(t)), \ i \in \{1, \ldots, N\}, \tag{3}$$

such that consensus is achieved if this error is equal to zero for all the agents, i.e., $\chi_i = \chi_j$ for all $i,j \in \mathcal{E}(\mathcal{G})$, $i \neq j$. The individual consensus error (3) can be expressed in matrix form as

$$e_c(t) = [e_{c_1}(t), e_{c_2}(t), \ldots, e_{c_N}(t)]^T = -(L \otimes I_n)\chi(t) \in \mathbb{R}^{nN}, \tag{4}$$

where $L \otimes I_n$ represents the Kronecker product between L and the identity matrix of size n.

2.2. Task-based control approach

Let us consider one agent of the form (1) with state vector $q \in \mathbb{R}^n$. Any task to be performed by the agent is defined by a differential mapping, denoted by $x(q)$, between a task space and the state space. To reach a desired value x^d for the task, an error function is defined as

$$e(q) = x(q) - x^d, \tag{5}$$

where $e(q), x(q)$ and $x^d \in \mathbb{R}^m$, with m the dimension of the task space. The time-derivative of (5) is given by

$$\dot{e} = J(q)\dot{q}, \tag{6}$$

being $J(q) \in \mathbb{R}^{m \times n}$ a Jacobian matrix. According to (1), $\dot{q}$ is equal to the vector of control inputs, and to solve for $\dot{q}$, the Moore-Penrose pseudoinverse of the Jacobian matrix can be used as defined in [34]

$$\dot{q} = J(q)^+ \dot{e}, \tag{7}$$

where $J(q)^+ = J(q)^T (J(q)J(q)^T)^{-1} \in \mathbb{R}^{n \times m}$. This approach is valid if the dimension of the task space is less than or equal to the dimension of the state space of the system ([34]), i.e. according to our notation, it must be accomplished that $m \leq n$. In the case of study presented in this work for UAVs moving in 3D, the degrees of freedom of each UAV are equal to 3 ($n = 3$), and the dimension of the tasks to be solved are 1 for obstacle avoidance ($m = 1$) and 3 for agents' formation ($m = 3$). Then, both task can be combined under the task-based control approach.

In (7), the error dynamics is a design choice that assigns a desired dynamics of the error function in order to guarantee its convergence, for instance

$$\dot{e} = -\lambda e, \tag{8}$$

with $\lambda > 0$, which yields exponential convergence of the task function to its desired value. Thus, the vector of control inputs is given by

$$\dot{q} = -\lambda J(q)^+ e(q). \tag{9}$$

2.3. Hierarchical task-based formulation

In the context that a single system or a MAS must perform several tasks simultaneously, a trade-off between all the tasks must be established, since the completion of higher priority tasks must be guaranteed, while others could be partially performed.

A solution to this problem was introduced in [27] as a hierarchical task-based approach and it has been applied to robotic systems ([28, 35, 36]). This approach assigns a fixed priority to each task and the one with the highest priority is named the priority task. The solution of the priority task is always used in the computation of the control input. The solutions of the other tasks are projected into the null space of the priority task avoiding contradictory solutions. Thus, lower priority tasks do not influence the solution of the priority task.

Following [35], the null space of the i-th task, considered the priority task, is computed as

$$N_i = I_n - J_i^+(q)J_i(q), \tag{10}$$

where $N_i \in \mathbb{R}^{n\times n}$. For instance, considering two tasks x_1 and x_2, where x_1 has the highest priority, the total control action is computed as ([28, 35])

$$\dot{q} = \dot{q}_1 + N_1\dot{q}_2, \tag{11}$$

where $\dot{q} \in \mathbb{R}^n$ is the control input of the system, $\dot{q}_1 = \lambda_1 J_1(q)^+ e_1(q) \in \mathbb{R}^n$ is the input computed from x_1 and $\dot{q}_2 = \lambda_2 J_2(q)^+ e_2(q) \in \mathbb{R}^n$ is the input computed from x_2. When only one task is active, a control input of the form (9) is applied. Otherwise, when a second task is activated, the control law changes to (11) and discontinuities appear in the control inputs due to the instantaneous switching between control laws when the tasks are activated or deactivated. This undesired effect can be avoided using the following control law ([36])

$$\dot{q} = \dot{q}_1' + \dot{q}_{1|2}, \tag{12}$$

with

$$\begin{aligned} \dot{q}_1' &= J_1^+ \dot{e}_1', \\ \dot{q}_{1|2} &= (J_2 N_1)^+ (\dot{e}_2 - J_2 J_1^+ \dot{e}_1'), \\ \dot{e}_1' &= h(t)\dot{e}_1 + (1 - h(t)) J_1 J_2^+ \dot{e}_2, \end{aligned}$$

where $\dot{e}_i$ assigns the desired dynamics (8) and $h(t)$ is a smooth scalar time-dependent function varying continuously from 0 to 1. It can be verified that the single control law (9) for the secondary task is obtained from (12) when $h(t) = 0$. This allows that the highest priority task starts to have effect in a smooth way and it is fully considered when $h(t) = 1$. This approach provides a continuous solution for $\dot{q}$ and it will be used in this work.

3. A solution to the problem of formation with collision avoidance

In this section we propose a solution to the following problem.

Definition 3.1. Let $\mathcal{A} = \{A_1, ..., A_N\}$ be a set of $N < \infty$ mobile agents, in particular UAVs, and $\mathbf{F}$ be the required formation of the agents given by a vector of relative distances of each agent with respect to an arbitrary common reference frame, and let us consider that there also exists a set of obstacles in the environment. The agents themselves can be obstacles for each other or there may be fixed obstacles in the environment. The formation with obstacle avoidance problem (FOAP) consists in finding control inputs yielding a trajectory for each agent such that the formation is reached and the trajectories avoid the obstacles.

The following assumptions are considered:

- Each agent has omnidirectional sensing capability and focuses to avoid only the nearest obstacle, i.e., one obstacle is considered at each time-instant,
- the communication link between agents is modeled by a Laplacian matrix for the whole MAS,

- the agents, together with the communication links, describe a connected graph (as defined in Section 2),

- each agent only has access to the position of their neighbors through the communication link for formation control purposes, and the position of the agents that are not neighbors is obtained from the omnidirectional sensor for collision avoidance purposes.

According to the FOAP definition, two tasks are needed, one to reach the desired formation and the other to avoid collisions. Since the agents must be driven to reach the formation avoiding any obstacle, then the priority task (herein named local task) is the collision avoidance, which is performed by each agent independently, and the secondary task (herein named global task) is the formation based on consensus of the agents' states.

The following notation is used during the solution of FOAP: Let $A_i \in \mathcal{A}$, then x_1^i and x_2^i are the local and global tasks associated to A_i. Also, $q_i = [x_i, y_i, z_i]$, $\dot{q}_i = [\dot{x}_i, \dot{y}_i, \dot{z}_i]$ are the position and velocity of the UAV A_i, and $q_o^i \in \mathbb{R}^3$ is the position of the nearest obstacle to agent A_i.

3.1. Local task (collision avoidance)

Every agent A_i must avoid collisions with obstacles (other agents or fixed obstacles) and to do that, the agents must always maintain a security distance (R) to the obstacles. In order to meet this objective, the task $x_1^i = \rho(q_i)$ is defined as the relative distance from the UAV position q_i to the position q_o^i of the nearest fixed obstacle as

$$\rho(q_i) = \|q_i - q_o^i\| \in \mathbb{R}. \tag{13}$$

Notice that the distance $\rho(q_i)$ can be measured with a sensor onboard the UAV A_i and thus, the global position of the obstacle is not required. Thus, the following error function is defined

$$e_{o_i} = \rho(q_i) - R \in \mathbb{R}. \tag{14}$$

The error dynamics is

$$\dot{e}_{o_i} = \frac{(q_i - q_o^i)^T}{\|q_i - q_o^i\|}\dot{q}_i \in \mathbb{R}. \tag{15}$$

In the case that the obstacle to be avoided is another UAV, we consider a different measure of relative distance as a security region ($\rho(q_i)$) modeled as a prolate ellipsoid in order to increase the separation distance in the vertical axis and keep the transverse distance, reducing the effect of the turbulence generated by the UAV's propellers. This provides a larger separation of the UAVs when one is below another. So, the relative distance from the agent A_i to the agent A_j is

$$\rho(q_i) = \sqrt{\frac{(x_i - x_j)^2}{a^2} + \frac{(y_i - y_j)^2}{b^2} + \frac{(z_i - z_j)^2}{c^2} - 1}, \tag{16}$$

where $a, b, c \in \mathbb{R}$ are the lengths of the semi-axes of the ellipsoid that represents the relative safety distance of each UAV and $q_o^i = [x_j, y_j, z_j]^T$ is the position of the nearest obstacle to the UAV. In this work, the parameters a, b represent the size of the 3D UAV, while parameter c represents the safety distance from the top of agent A_i to the bottom of agent A_j to avoid the turbulence generated by A_i over A_j. According to (16), the error dynamics is

$$\dot{e}_{o_i} = \frac{(E(q_i - q_o^i))^T}{\rho(q_i)}\dot{q}_i \in \mathbb{R}, \tag{17}$$

where

$$E = \begin{bmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{bmatrix}.$$

Depending on the nearest obstacle, the error dynamics can be (15) when the obstacle is fixed or (17) when the obstacle is another UAV. This strategy assumes that as both UAVs in potential collision will realize the same strategy, they will made quasistatic motion during the collision avoidance. In general, the error dynamics is represented by

$$\dot{e}_{o_i} = J_{o_i}(q)\dot{q}_i. \tag{18}$$

Establishing desired dynamics for (18) as $\dot{e}_{o_i} = -\lambda e_{o_i}$ and solving for $\dot{q}_i$, the following equation is obtained

$$\dot{q}_i = -\lambda J_{o_i}^{+}(q) e_{o_i}. \tag{19}$$

3.2. Global task (agents' formation)

As mentioned in the Introduction Section, the formation problem is addressed in this work as a consensus problem to formulate a distributed solution. Consider a set of N UAVs connected through a communication network such that they exchange information with each other. We will assume single integrator dynamics of each axis of motion for each UAV ([33])

$$\dot{q}_i = u_i(t) \quad i \in 1, ..., N, \tag{20}$$

where $q_i \in \mathbb{R}^n$ is the vector of the i-th UAV's position and $u_i \in \mathbb{R}^n$ is the control input given by its linear velocities. Agents' formation can be achieved by consensus of a virtual network. For instance, in [22], agents' formation is specified as a set of fixed translation vectors $v_i \in \mathbb{R}^n$ with respect to an arbitrary common reference frame, thus the position q_i of the i-th agent is related to the position q_{v_i} of the virtual agent by

$$q_{v_i} = q_i + v_i, \tag{21}$$

where $q_{v_i} \in \mathbb{R}^n, i \in 1, ..., N$. Notice that the network of virtual agents has the same Laplacian matrix than the original network and the virtual agent's dynamics is given by

$$\dot{q}_{v_i} = \dot{q}_i = u_i(t). \tag{22}$$

The consensus error for the virtual agent i with respect to its neighbors is

$$e_{v_i} = \sum_{j \in \mathcal{N}_i} (a_{ij}(q_{v_j} - q_{v_i})) \in \mathbb{R}^n, \tag{23}$$

and the consensus error vector is

$$e_v = [e_{v_1}, e_{v_2}, ..., e_{v_N}] = -(L \otimes I_n) q_v(t) \in \mathbb{R}^{nN}. \tag{24}$$

Applying adequate control inputs to the systems (20) in terms of (24), the virtual agents will reach consensus and consequently the real agents will reach the desired formation specified by the vectors v_i. It is worth noting that the consensus error of each agent in (23) only depends on the position of its neighbors. Thus, each agent will compute independently its control law in terms of this local consensus, which represents a distributed setup. In the following subsections, we propose control laws to accomplish this global task.

3.2.1. Finite-time control law

The following control protocol is proposed to solve the consensus of the virtual agents and consequently the real agents will reach the desired formation:

$$u(t) = \dot{q}_v = k\lfloor e_v \rceil^{\frac{1}{2}} = k \begin{bmatrix} \lfloor e_{v_1} \rceil^{\frac{1}{2}} \\ \vdots \\ \lfloor e_{v_N} \rceil^{\frac{1}{2}} \end{bmatrix} \in \mathbb{R}^{nN}, \tag{25}$$

where

$$\lfloor e_{v_i} \rceil^{1/2} \quad = \quad \left[\lfloor e_{v_i(x)} \rceil^{1/2}, \lfloor e_{v_i(y)} \rceil^{1/2}, \lfloor e_{v_i(z)} \rceil^{1/2} \right]^T \in \mathbb{R}^n, \tag{26}$$

considering that $\lfloor \bullet \rceil^{\frac{1}{2}} = |\bullet|^{\frac{1}{2}} \text{sign}(\bullet)$. The next proposition states that the proposed control law achieves consensus (hence also the desired formation) in finite time.

Proposition 3.1. *Consider a MAS modeled as a connected undirected graph $\mathcal{G}$ and the vector of consensus error e_v given in (24). Then, there exists a control gain $k \in \mathbb{R}^+$ such that the following nonlinear distributed control protocol*

$$u_i(t) = \dot{q}_{v_i} = k\lfloor e_{v_i} \rceil^{\frac{1}{2}} \in \mathbb{R}^n, \tag{27}$$

achieves finite-time convergence to zero of the consensus error vector (24) *and consequently consensus of the state of the virtual agents' system* (22)*, from any initial state $q_v(0)$.*

Proof. Considering the consensus error dynamics of the virtual agents

$$\dot{e}_v = -kL\Phi(e_v), \tag{28}$$

where $\Phi(e_v)$ is given by

$$\Phi(e_v) = \begin{bmatrix} \lfloor e_{v_1} \rceil^{\frac{1}{2}} \\ \vdots \\ \lfloor e_{v_N} \rceil^{\frac{1}{2}} \end{bmatrix}, \tag{29}$$

and let

$$V(e) = \sum_{i=1}^{N} |e_{v_i}| \tag{30}$$

be a candidate Lyapunov function for (28), which is positive definite but not continuously differentiable for all e_v. However, since (30) is Lipschitz continuous, the global stability of (28) is obtained if $\dot{V}$ is negative definite almost everywhere [37, p. 207], which will be demonstrated in the sequel. To this aim, let $S(e_v) = [\text{sign}(e_{v_1}) \ \cdots \ \text{sign}(e_{v_N})]^T$ and notice that if $e_{v_i} \neq 0$ then the i-th element of $S(e_v)^T L$ is either zero or it has the same sign as e_{v_i}. Then, $S(e_v)^T L\Phi(e_v) = S(e_v)^T W(t)\Phi(e_v)$ where $W(t) = diag(w_1(t), \ldots, w_N(t))$, $w_i(t) \geq 0$. Then

$$\dot{V} = -S(e_v)^T L\Phi(e_v) = -k\sum_{i=1}^{N} w_i(t)|e_{v_i}|^{\frac{1}{2}}. \tag{31}$$

Notice that, with $e_{v_i} \neq 0$, $w_i(t)$ is zero iff $\forall j \in \mathcal{N}_i(\mathcal{G})$, $\text{sign}(e_{v_i}) = \text{sign}(e_{v_j})$. Moreover, since $e_v = (L \otimes I_n)q_v = 0$ then along the evolution of the system it holds that $\mathbf{1}_N e_v^T = \sum e_{v_i} = 0$ and therefore, unless $e_v = 0$ there always exists a $e_{v_i} \neq 0$ with $w_i(t) \neq 0$, i.e. a node with nonzero consensus error such that $\exists j \in \mathcal{N}_i(\mathcal{G})$, $\text{sign}(e_{v_i}) \neq \text{sign}(e_{v_j})$. Thus, the origin of (28) is globally asymptotically stable. As we have $e_v = (L \otimes I_n)q_v = 0$ then $q_{v_1} = \ldots = q_{v_n}$ and consensus is achieved.

Since (28) is globally asymptotically stable and for all $\lambda > 0$, $\Phi(e_v) = \lambda^{-(d+1)}\Phi(\lambda e_v)$ with $d = -\frac{1}{2}$ (i.e. the vector field is homogeneous with negative degree with respect to the standard dilation), finite-time stability follows from [38, Theorem 7.1]. Thus, consensus is achieved in finite-time. □

3.2.2. Predefined-time control law

In this section, we present an alternative control law that unlike the one of (27), it is able to drive the MAS to the desired formation in a specified time. The control law is based on the tracking of a reference given by a time base generator (TBG), which is a parametric time signal that converges to zero in a specified time, as defined in [16].

Theorem 3.2 ([16])**.** Consider a MAS modeled as a connected undirected graph. Let $k_f \in \mathbb{R}^+$ be a constant gain, t_f a predefined-convergence time, and a TBG function $h_p(t)$ fulfilling the conditions $h_p(0) = 1$, $h_p(t \geq t_f) = 0$, $\dot{h}_p(0) = 0$ and $\dot{h}_p(t \geq t_f) = 0$. Then, the time-varying linear control protocol

$$u_i(t) = \dot{q}_{v_i} = -\dot{h}_p(t)e_{v_i}(0) + k_f(e_{v_i}(t) - h_p(t)e_{v_i}(0)) \in \mathbb{R}^n, \tag{32}$$

with $e_{v_i}(t)$ given by (23) and $e_{v_i}(0)$ computed from the initial state $q_v(0)$, achieves predefined-time convergence to zero of the consensus error vector (24) and consequently consensus of the state of the virtual agents' system (22), from any initial state $q_v(0)$.

Proof. See the proof of Theorem 7 in [16] for the complete procedure. A notion of the proof is as follows: the control law (32) can be written in vector notation as

$$u(t) = \dot{q}_v = -\dot{h}_p(t)e_v(0) + k_f\xi(t) \in \mathbb{R}^{nN}, \tag{33}$$

where

$$\xi(t) = e_v(t) - h_p(t)e_v(0) \in \mathbb{R}^{nN} \tag{34}$$

is the tracking error of the consensus error trajectory (e_v) with respect to the desired trajectory given by the TBG ($h_p(t)e_v(0)$). The vector of control inputs $u(t)$ introduced in (22) (expressed in vector notation) is then used to prove the convergence of the tracking error to zero. Since tracking of the TBG reference is guaranteed, $e_v(t)$ follows the profile $h_p(t)e_v(0)$. This profile initiates in $e_v(0)$ ($h_p(t) = 1$) and continuously converges to zero at time t_f ($h_p(t \geq t_f) = 0$). Therefore, predefined-time convergence to zero of the consensus error $e_v = (L \otimes$

$I_n)q_v = 0$ is achieved. Then, the state of the virtual agents reaches $q_{v_1} = \ldots = q_{v_n}$ at time t_f. □

The result presented in [7] ensures that the consensus of the virtual agents for both control laws (27) and (32) is the average of the initial conditions of the agents if the Laplacian matrix L comes from a strongly connected graph, i.e.

$$\lim_{t\to\infty} q_{v_i(x)}(t) = \alpha_x \in \mathbb{R}, \tag{35}$$

with

$$\alpha_x = \frac{\sum_i \gamma_i q_{v_i(x)}(0)}{\sum_i \gamma_i}, \tag{36}$$

where γ_i is the i-th element of the left eigenvector γ of L, $q_{v_i(x)}$ is the position of the virtual agent i in the x-axis and α_x represents the average of the initial conditions of the virtual agents in the x-axis. Similar expressions can be formulated for the other coordinates and then, the consensus vector for the 3D MAS is given by

$$\bar{q}_v = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}, \tag{37}$$

where α_y and $\alpha_z \in \mathbb{R}$ are the average of the initial conditions of the virtual agents for the y-axis and z-axis, respectively. The fact that we know that consensus is achieved for the virtual agents' state, allows in the hierarchical task-based approach the definition of the secondary task as $x_2^i = q_{v_i}$ (the position of the virtual agents) and the error of this task as

$$e_c = q_v - (\mathbf{1}_N \otimes q_v^*) \in \mathbb{R}^{nN}, \tag{38}$$

with $q_v^* \in \mathbb{R}^n$ being a constant vector that in the particular case in which only a consensus task is realized $q_v^* = \bar{q}_v$. Thus, the corresponding time-derivative of the consensus task is given by

$$\dot{e}_c = J_c(q)\dot{q}_v \tag{39}$$

where $J_c(q) = I_n \otimes J_{c_i}(q)$. Since $J_{c_i}(q) = I_n$, because the right-hand side term of (38) is constant, then $J_c(q) = I_{nN} \in \mathbb{R}^{nN \times nN}$ and it is the Jacobian matrix of the consensus task. In the sequel, we will remove the explicit dependence of the Jacobians from q. Recall that according to (22), $\dot{q}_v = \dot{q}$.

3.3. Hierarchical combination of global and local tasks

Following [36], the control law for each agent combining the obstacle avoidance as primary (local) task, denoted by subscript o, and UAVs formation as secondary (global) task, denoted by subscript c, is given by

$$\dot{q}_i = \dot{q}'_{o_i} + \dot{q}_{o|c_i}, \tag{40}$$

where:

$$\begin{aligned} \dot{q}'_{o_i} &= J^+_{o_i} \dot{e}'_{o_i}, \\ \dot{q}_{o|c_i} &= (J_{c_i} N_{o_i})^+ (\dot{e}_{c_i} - J_{c_i} J^+_{o_i} \dot{e}'_{o_i}), \\ \dot{e}'_{o_i} &= h(t) \dot{e}_{o_i} + (1 - h(t)) J_{o_i} J^+_{c_i} \dot{e}_{c_i}, \end{aligned}$$

and $0 \leq h(t) \leq 1$ is a smooth scalar function. The transition function $h(t)$ is activated increasing from 0 to 1 when an obstacle is detected within the safety distance. While the obstacle is within the safety distance, the value of $h(t) = 1$ and when the obstacle leaves the safety distance the function $h(t)$ is deactivated decreasing from 1 to 0. The transition function uses a finite-time for activation and deactivation, i.e., the time to carry out the obstacle avoidance task is bounded. Thus, the smoothness of $\dot{q}_i$ is achieved thanks to the use of the transition function.

In the following theorem, we will prove the stability of the control law including both collision avoidance and formation control tasks. In the following proof, both tasks errors will be expressed in vector form to treat the MAS as a single system, but keep in mind the distributed individual computation of the agents control law (40). Besides, we will specify the form of the two tasks vector dynamics $\dot{e}_o$ and $\dot{e}_c$ in closed-loop for the two alternatives of formation control protocols.

Theorem 3.3. *Consider a MAS of N UAVs modeled as n-order systems as in* (20)*, with a connected undirected communication topology. For this kind of MAS, the control law* $\dot{q} = \left[\dot{q}_1^T, \dot{q}_2^T, \cdots \dot{q}_N^T\right]^T$ *with* $\dot{q}_i$ *as in* (40) guarantees convergence to zero of the consensus error e_c (38) in finite-time if (25) is used or in predefined-time if (33) is used, and also convergence to zero of the obstacle avoidance error (e_o) in spite that both tasks are active with a transition function $0 \leq h(t) \leq 1$. The terms of (40) are

$$
\begin{aligned}
\dot{e}_o &= -\lambda e_o \in \mathbb{R}^N, \text{ with } e_o = [e_{o_1}, e_{o_2}, \cdots e_{o_N}]^T \text{ and } e_{o_i} \in \mathbb{R} \text{ as in (14)},\\
J_{c_i} &= I_n, \text{ according to (39)},\\
(J_c N_o)^+ &= \begin{bmatrix} (J_{c_1} N_{o_1})^+ & 0 & \cdots & 0 \\ 0 & (J_{c_2} N_{o_2})^+ & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (J_{c_N} N_{o_N})^+ \end{bmatrix},\\
J_o^+ &= \begin{bmatrix} J_{o_1}^+ & 0 & \cdots & 0 \\ 0 & J_{o_2}^+ & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_{o_N}^+ \end{bmatrix}, N_o = \begin{bmatrix} N_{o_1} & 0 & \cdots & 0 \\ 0 & N_{o_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_{o_N} \end{bmatrix},
\end{aligned}
$$

where $J_{o_i} \in \mathbb{R}^{n\times 1}$ *is given by* (18)*,* $N_{o_i} = I_n - J_{o_i}^+ J_{o_i} \in \mathbb{R}^{n\times n}$*, and the term* $\dot{e}_c$ *is given by* (25) *to achieve consensus in finite-time or* (33) *for consensus in predefined-time.*

Proof. The proof will be presented for each control law (25) and (33).

- *Finite-time control law* (25). In this case, let us propose the following extended error as

$$
e' = \begin{bmatrix} e_c \\ e_o \end{bmatrix}. \tag{41}
$$

Then, consider the Lyapunov candidate function

$$
V = \frac{1}{2} e'^T e', \tag{42}
$$

with time derivative

$$\dot{V} = e'^T \dot{e}'. \tag{43}$$

To compute $\dot{V}$, we first need the open-loop dynamics of $\dot{e}_c$ and $\dot{e}_o$ for the whole network. On the one hand, the dynamics $\dot{e}_c$ are given by (39). On the other hand, expanding the obstacle task error for the N agents, we get

$$\dot{e}_o = J_o \dot{q} = \begin{bmatrix} J_{o_1} & 0 & \cdots & 0 \\ 0 & J_{o_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_{o_N} \end{bmatrix} \dot{q}, \tag{44}$$

where $J_o \in \mathbb{R}^{N \times nN}$. Now we have:

$$\dot{V} = \begin{bmatrix} e_c^T & e_o^T \end{bmatrix} \begin{bmatrix} J_c \\ J_o \end{bmatrix} \dot{q}. \tag{45}$$

Introducing the hierarchical task-based control protocol (40), then

$$\dot{V} = \begin{bmatrix} e_c^T & e_o^T \end{bmatrix} \Gamma, \tag{46}$$

where

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} = \begin{bmatrix} J_c(J_o^+ \dot{e}_1') + J_c((J_c N_o)^+ (\dot{e}_c - J_{ce} J_o^+ \dot{e}_1')) \\ J_o(J_o^+ \dot{e}_1') + J_o((J_c N_o)^+ (\dot{e}_c - J_c J_o^+ \dot{e}_1')) \end{bmatrix}.$$

Expanding the expressions using $\dot{e}_1'$ as in (40), we have

$$\begin{aligned} \Gamma_1 = {} & J_c J_o^+ \left[h(t)\dot{e}_o + (1-h(t)) J_o J_c^+ \dot{e}_c \right] \\ & + J_c (J_c N_o)^+ \left\{ \dot{e}_c - J_c J_o^+ \left[h(t)\dot{e}_o + (1-h(t)) J_o J_c^+ \dot{e}_c \right] \right\}, \end{aligned} \tag{47}$$

$$\begin{aligned} \Gamma_2 = {} & J_o J_o^+ \left[h(t)\dot{e}_o + (1-h(t)) J_o J_c^+ \dot{e}_c \right] \\ & + J_o (J_c N_o)^+ \left\{ \dot{e}_c - J_c J_o^+ \left[h(t)\dot{e}_o + (1-h(t)) J_o J_c^+ \dot{e}_c \right] \right\}. \end{aligned} \tag{48}$$

According to the properties $J_o J_o^+ = I_N$, $J_o N_o = 0_{N \times nN}$, $N_o = N_o^T$, and $N_o N_o = N_o$, the following holds

$$J_o (J_c N_o)^+ = J_o N_o^T (N_o N_o^T)^{-1} = J_o N_o (N_o N_o^T)^{-1} = 0_{N \times nN}, \tag{49}$$

$$(J_c N_o)^+ = (J_c N_o)^T ((J_c N_o)(J_c N_o)^T)^{-1} = N_o (N_o N_o^T)^{-1} = I_{nN}. \tag{50}$$

Thus, Γ_1 and Γ_2 are simplified as follows:

$$\begin{aligned} \Gamma_1 &= \dot{e}_c, \\ \Gamma_2 &= (1-h(t))J_o\dot{e}_c + h(t)\dot{e}_o. \end{aligned} \tag{51}$$

In these expressions, we need to introduce the closed-loop dynamics $\dot{e}_o = -\lambda e_o$ for collision avoidance and $\dot{e}_c = k\lfloor e_v \rceil^{\frac{1}{2}}$ from (25) for formation control. For the last one, since

$$|e_{v_i(x)}|^{1/2}\text{sign}(e_{v_i(x)}) = \frac{|e_{v_i(x)}|}{|e_{v_i(x)}|^{1/2}}\text{sign}(e_{v_i(x)}) = \frac{e_{v_i(x)}}{|e_{v_i(x)}|^{1/2}},$$

and similarly for the y- and z-axes. According to (26), we have that $\lfloor e_{v_i} \rceil^{\frac{1}{2}} = \Delta(e_{v_i})e_{v_i}$, where

$$\Delta(e_{v_i}) = \text{diag}\left\{|e_{v_i(x)}|^{-1/2}, |e_{v_i(y)}|^{-1/2}, |e_{v_i(z)}|^{-1/2}\right\} \in \mathbb{R}^{n\times n}. \tag{52}$$

Now, considering all the agents A_i

$$\Delta := \Delta(e_v) = \text{diag}\left\{\Delta(e_{v_1}), \ldots, \Delta(e_{v_N})\right\} \in \mathbb{R}^{nN\times nN}. \tag{53}$$

Furthermore, according to the properties of the Laplacian matrix, from (24) and (38), we have

$$e_v = -(L \otimes I_n)e_c, \tag{54}$$

therefore

$$\dot{e}_c = -k\Delta(L \otimes I_n)e_c. \tag{55}$$

Introducing (55) and $\dot{e}_o = -\lambda e_o$ in (51), the time derivative of the Lyapunov function is

$$\dot{V} = -\begin{bmatrix} e_c^T & e_o^T \end{bmatrix} M \begin{bmatrix} e_c \\ e_o \end{bmatrix}, \tag{56}$$

where

$$M = \begin{bmatrix} k\Delta(L \otimes I_n) & 0_{N\times N} \\ k(1-h(t))J_o\Delta(L \otimes I_n) & \lambda h(t) I_N \end{bmatrix}. \tag{57}$$

The eigenvalues of matrix M depend on the constants values $k > 0$, $\lambda > 0$, and on $h(t)$ and matrices L and Δ. The matrix L represents a connected and

balanced graph, it has an eigenvalue $\lambda_1(L) = 0$ with an associated eigenvector $\mathbf{1} = [1 \cdots 1]^T$ such that $L\mathbf{1}^T = 0$ for each coordinate (x, y, z) of the consensus error, which implies that $e_{c_1} = \cdots = e_{c_N}$. Moreover, if the matrix ΔL is not balanced, but L has a right eigenvector γ associated with $\lambda_1(L) = 0$ that satisfies $L\gamma = 0$, therefore $\Delta L\gamma = 0$ with $\gamma = [\gamma_1, \ldots, \gamma_N]$ and $\gamma_1 = \cdots = \gamma_N$. Then $e_c = \gamma$, which means that consensus is achieved. Since $\lambda > 0$ and $0 \leq h(t) \leq 1$, then the matrix M is positive semidefinite. Thus, $\dot{V} < 0$ for $t < t_f$ and $\dot{V} = 0$ when consensus is achieved and one obstacle is present with $h(t) > 0$. Furthermore, e_o converges asymptotically when an obstacle is present and e_c converges to a consensus value. Moreover, by the control law (25) the consensus is achieved in a finite time according to Proposition 3.1, i.e. that $\lim_{t \to t_f} k e_c^T \Delta (L \otimes I_n) e_c = 0$, where t_f is a finite convergence time. Besides, if $e_c = \gamma$ then $e_v = 0$ by (54), and since $e_v = -(L \otimes I_n) q_v = 0$, then $q_{v_1} = \ldots = q_{v_n}$ and consensus of the virtual agents positions is achieved, hence the formation of the real UAVs.

• *Predefined-time control law* (33). In this case, let us consider the following extended error

$$e' = \begin{bmatrix} \xi \\ e_o \end{bmatrix}, \tag{58}$$

where ξ is defined in (34) as $\xi(t) = e_v(t) - h_p(t) e_v(0)$. Let us use the same Lyapunov candidate function (42). In this case, its time derivative is given by:

$$\dot{V} = e'^T \dot{e}' = \begin{bmatrix} \xi^T & e_o^T \end{bmatrix} \begin{bmatrix} \dot{\xi} \\ \dot{e}_o \end{bmatrix}. \tag{59}$$

Given the definition of ξ and according to (54) and (38), where $e_v = -(L \otimes I_n) e_c = -(L \otimes I_n) q_v$, the time derivative of the tracking error between the consensus error and the TBG reference is as follows

$$\dot{\xi} = \dot{e}_v(t) - \dot{h}_p(t) e_v(0) = -(L \otimes I_n)\dot{q} + \dot{h}_p(t) e_v(0). \tag{60}$$

Substituting (60) and (44) in (59), we have

$$\dot{V} = \begin{bmatrix} \xi^T & e_o^T \end{bmatrix} \begin{bmatrix} -(L \otimes I_n) \\ J_o \end{bmatrix} \dot{q} + \dot{h}_p(t) \xi^T e_v(0). \tag{61}$$

Introducing the hierarchical task-based control protocol (40), then

$$\dot{V} = \begin{bmatrix} \xi^T & e_o^T \end{bmatrix} \Gamma_p + \dot{h}_p(t) \xi^T e_v(0), \tag{62}$$

where

$$\Gamma_p = \begin{bmatrix} \Gamma_{p1} \\ \Gamma_{p2} \end{bmatrix} = \begin{bmatrix} -(L \otimes I_n)(J_o^+ \dot{e}_1') - (L \otimes I_n)((J_c N_o)^+ (\dot{e}_c - J_c J_o^+ \dot{e}_1')) \\ J_o(J_o^+ \dot{e}_1') + J_o((J_c N_o)^+ (\dot{e}_c - J_c J_o^+ \dot{e}_1')) \end{bmatrix}. \tag{63}$$

By a similar procedure as in the finite-time approach and using the properties of (49), the simplified matrix Γ_p is

$$\Gamma_p = \begin{bmatrix} \Gamma_{p_1} \\ \Gamma_{p_2} \end{bmatrix} = \begin{bmatrix} (L \otimes I_n)\dot{e}_c \\ (1 - h(t))J_o \dot{e}_c + h(t)\dot{e}_o \end{bmatrix}. \tag{64}$$

Using in these expressions the closed-loop dynamics $\dot{e}_o = -\lambda e_o$ for collision avoidance and $\dot{e}_c = -\dot{h}_p(t) e_v(0) + k_f \xi$ (33) for consensus of the virtual agents, (61) can be written as

$$\dot{V} = -\begin{bmatrix} \xi^T & e_o^T \end{bmatrix} M_p \begin{bmatrix} \xi \\ e_o \end{bmatrix} + \delta_p(t), \tag{65}$$

where

$$M_p = \begin{bmatrix} k_f(L \otimes I_n) & 0_{N \times N} \\ (1 - h(t))k_f J_o (L \otimes I_n) & \lambda h(t) I_N \end{bmatrix}, \tag{66}$$

$$\delta_p(t) = \dot{h}_p(t) \left(\xi^T \left(e_v(0) - (L \otimes I_n) e_v(0) \right) - (1 - h(t)) e_o^T J_o (L \otimes I_n) e_v(0) \right).$$

On the one hand, the term $\delta_p(t)$ can be seen as a disturbance that becomes null after time t_f, due that it depends on the time derivative of the function $h_p(t)$ that fulfills $\dot{h}_p(0) = 0$ and $\dot{h}_p(t \geq t_f) = 0$. On the other hand, the eigenvalues of matrix M_p depend on the constants values $k_f > 0$, $\lambda > 0$, and on $h(t)$ and matrix L. The matrix L represents a connected and balanced graph, it has an eigenvalue $\lambda_1(L) = 0$ with an associated eigenvector $\mathbf{1} = [1 \cdots 1]^T$ such that $L\mathbf{1}^T = 0$, for each coordinate (x, y, z) of the consensus error, which implies that $\xi_1 = \cdots = \xi_n$. We consider that $e_v(0)$ is known and consequently the initial tracking error is zero (since $h_p(0) = 1$). It is proved in [16] that there exists k_f

such that the tracking error converges to the origin although the initial value of $e_v(0)$ was uncertain. This implies that e_v tracks the reference $h_p(t)e_v(0)$. Thus e_v converges to the origin at the predefined time t_f at which $h_p(t)$ becomes null and then consensus is achieved. Since $\lambda > 0$ and $0 \leq h(t) \leq 1$, the matrix M_p is positive semidefinite and thus, $\dot{V} < 0$ for $t < t_f$ and $\dot{V} = 0$ when consensus is achieved and one obstacle is present with $h(t) > 0$. Furthermore, the e_o converges asymptotically when there is an obstacle and e_v converges to the origin around the predefined time using the control law (33). The predefined time to achieve consensus of the virtual agents might be affected if the task to avoid the obstacles is activated when e_v is near to the origin, but it is guaranteed that the origin will be reached. Moreover, when $e_v = 0$ and since $e_v = -(L \otimes I_n)q_v$, then $q_{v_1} = \cdots = q_{v_n}$ and consensus of the virtual agents positions is achieved, hence the formation of the real UAVs. □

Remark 1. The previous theorem guarantees that the consensus error of the virtual agents converges to the origin and consequently the desired formation is achieved in finite or predefined time in the absence of obstacles. In the presence of a finite number of obstacles and obstacles with a finite size, the desired formation is achieved in finite time in spite of the activation and deactivation of the obstacle avoidance task for some agents. This is due that the predefined time in which the desired formation is achieved by the MAS might be affected by the time to avoid obstacles if the collision avoidance task must be performed close to achieve the final formation. Nevertheless, since the global position of the formation is not prefixed, the agents adapt its trajectory to reach the formation in finite time although they perform an obstacle avoidance task at the end.

Remark 2. As stated at the beginning of Section 3, it is assumed that each agent has omnidirectional sensing capability and focuses to avoid only the nearest obstacle. Thus, the obstacles can be of irregular shapes and when an obstacle is detected by a UAV, the distance to the nearest point over the obstacle is obtained and the obstacle avoidance task is activated. The UAV performs a motion according to the control law combining the formation and obstacle avoidance

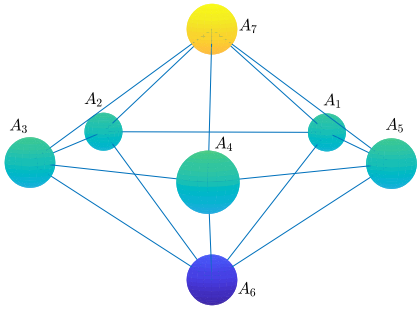
goals, and then, the nearest point to the obstacle is updated at each iteration until the obstacle is not detected. In this way, the proposed approach is valid to avoid obstacles with irregular shapes.

Remark 3. Notice that if we consider dynamic networks switching among connected graphs, the dynamics of the network becomes a switched system. In [14], using an approach to find a common Lyapunov function [39] independent of the communication topology, it was shown that the protocol (27) is a finite-time consensus algorithm for dynamic networks. A similar analysis may be done for the predefined-time control law (32) and both proposed protocols work for changing communication topology. However, the time-varying topology introduces an undesired effect of discontinuities in the control inputs when the consensus error changes due to the topology. Since our interest is to provide a reference signal to be tracked by the UAVs, it is required to ensure continuous velocities. In the case of dynamic networks, continuous control signals can be generated by using dynamic extension, i.e. increasing the order of the integrator chain. This extension is considered as future work.

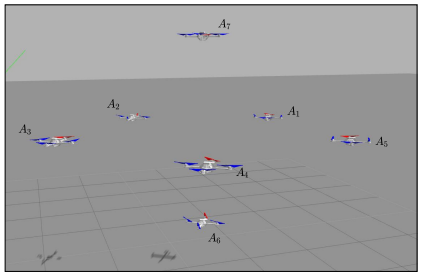
4. Simulation results

In this section, the proposed approach is evaluated in a virtual environment using the dynamic simulator Gazebo with a group of UAVs and all the functionalities programmed in ROS. The goal is to achieve the formation of the agents represented in Fig. 1 and at the same time avoiding collisions among agents and fixed obstacles in the environment. As a simulation example, we consider a seven-agent system ($N = 7$) in a network with undirected communication topology described by the graph of Fig. 1a.

In the simulation, we set $a_{ij} = 1$ and initial conditions $q_x(0) = [5, 3, 3, 5, 6, 7, 5]^T$, $q_y(0) = [-1, -1, 7, 5, 1, 3, 2]^T$ and $q_z(0) = [1.2, 1.2, 4, 1.5, 2, 2.5, 3]^T$ for agents 1 to 7, respectively. There are 5 columns in the environment as fixed obstacles in positions $q_{o_1} = [4, 6]$, $q_{o_2} = [6, 2]$, $q_{o_3} = [2, 0]$, $q_{o_4} = [2, 3]$ and $q_{o_5} = [6, 7]$, and the security distance between obstacles and agents was set



(a) Connected graph used in the simulations



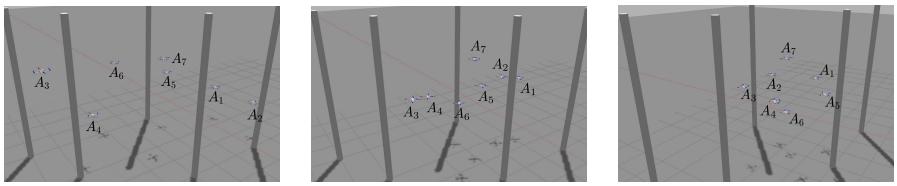
(b) Specified formation in the dynamic simulator Gazebo

Figure 1: Communication graph and desired formation of the UAVs.

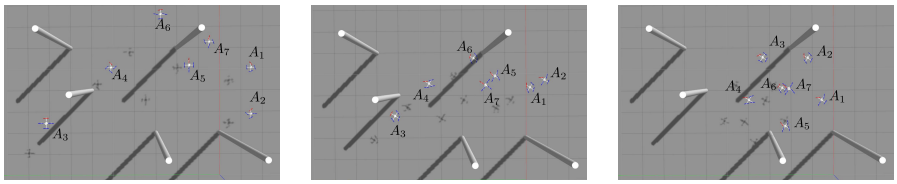
to $R = 0.5$. The displacement vectors for the virtual agents from A_1 to A_5 are $v_i = [1.5\cos((72i)^\circ), 1.5\sin((72i)^\circ), 0]\,, i \in 1, ..., 5$ with respect to the formation center for each agent respectively, and for virtual agents A_6 and A_7 are $v_6 = [0, 0, -1]$ and $v_7 = [0, 0, 1]$. In the implementation, the transition function is $h(t) = \frac{1}{2}\left(1 - \cos\left(\pi \frac{(t-t_0)}{(t_f - t_0)}\right)\right)$, where t_0 is set to the current time value $(t_0 = t)$ at the instant that a UAV crosses the security distance of an obstacle, $t_f = t_0 + t_d$ with t_d the duration of the transition function, in the implementation $t_d = 0.2s$, which represents the time that takes the UAV to carry out the obstacle avoidance.

4.1. Finite-time convergence

In this subsection, the results of the proposed scheme detailed in Theorem 3.3 are presented using the finite-time consensus control law (25) with $k = 0.5$. The results of the dynamic simulation for the two tasks of the UAVs (consensus to achieve a formation and obstacle avoidance) are presented in Figs. 2-5. Fig. 2 shows snapshots of the UAVs motion in the environment with obstacles from two different perspectives (isometric views in the first row, and upper views in the second row). The images on the left correspond to the initial position of the UAVs; the images at the center present an intermediate position during the UAVs motion to get the formation, and the images on the right show that the desired formation was effectively reached. It is worth noting that some of the

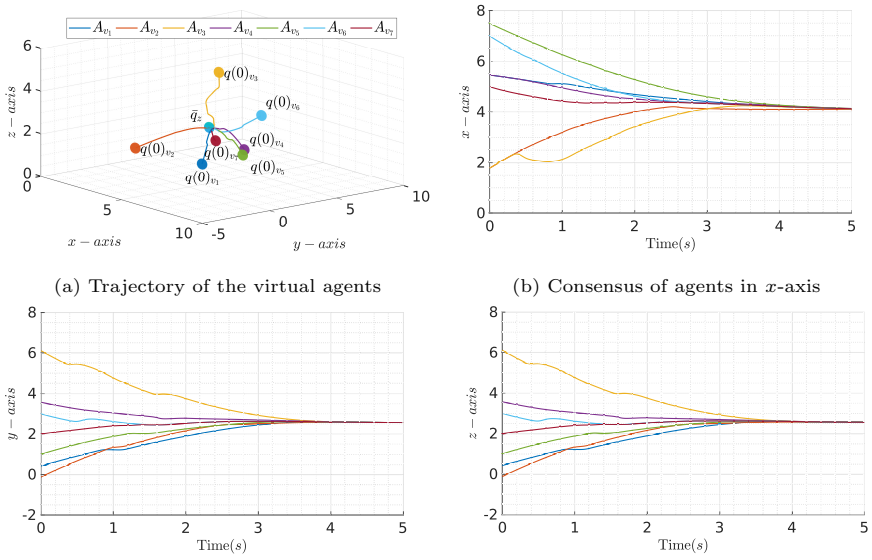


(a) Initial position (b) Intermediate position (c) Final position



(d) Upper view of initial position (e) Upper view of intermediate position (f) Upper view of final position

Figure 2: Snapshots of the UAVs motion in the simulation.

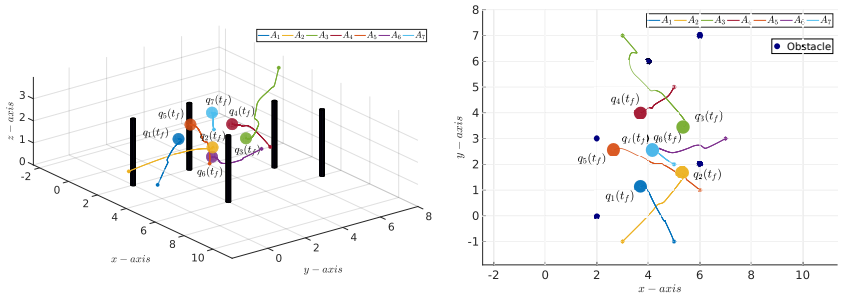


(a) Trajectory of the virtual agents (b) Consensus of agents in x-axis

(c) Consensus of agents in y-axis (d) Consensus of agents in z-axis

Figure 3: **Finite-time convergence.** Consensus of virtual agents with convergence in finite-time around 5s.

UAVs had to avoid obstacles during their motion (for instance, see A_6 in Fig. 2e) and this aspect will be clearer in the subsequent figures of this subsection.



(a) Isometric view (b) Upper view

Figure 4: **Finite-time convergence.** Trajectory of the UAVs reaching formation and avoiding obstacles.

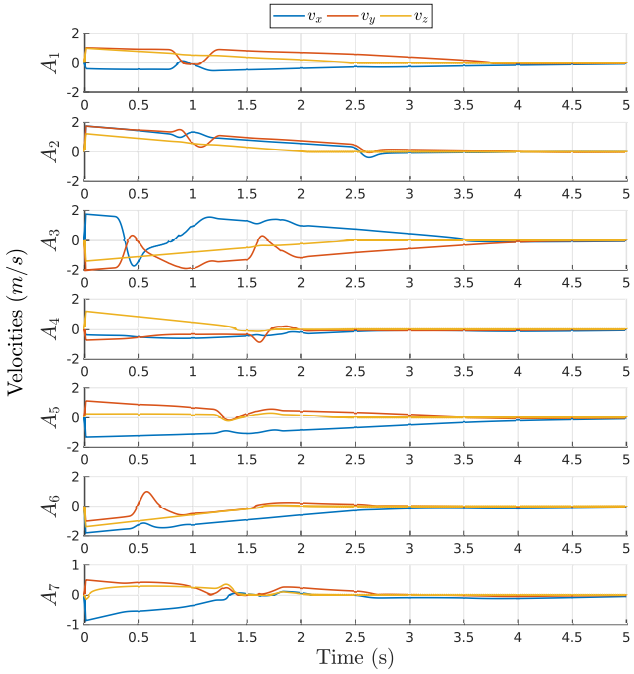


Figure 5: **Finite-time convergence.** Control inputs (velocities) of the UAVs.

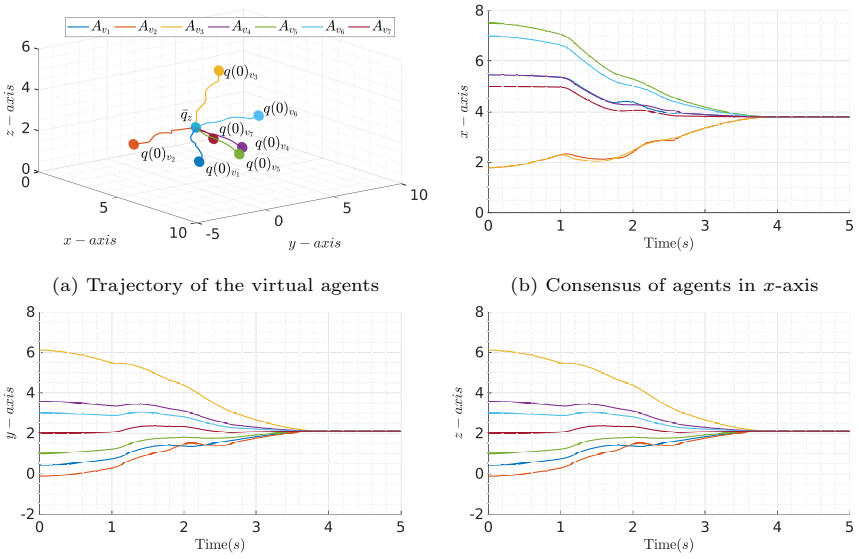
Fig. 3 shows the virtual agents motion to achieve consensus. The top-

left subfigure (Fig. 3a) depicts the 3D trajectory of the virtual agents, i.e., the evolution of each q_{v_i}. It can be seen that consensus in these variables is achieved by reaching a common value q_v^*. The other subfigures show the evolution in time of the virtual agents position for each coordinate with consensus values $q_{v_x}^* = 4.13$ (Fig.3b), $q_{v_y}^* = 2.56$ (Fig. 3c) and $q_{v_z}^* = 2.93$ (Fig. 3d). Then, consensus of the virtual agents is achieved in around 5 seconds using $k = 0.5$ as control gain in the finite-time consensus control law (25). Faster convergence could be achieved using a larger control gain, however, the convergence time would be different for other initial positions of the UAVs. Fig. 4 presents the trajectories of the UAVs in 3D from their initial positions (marked with an asterisk) and the obstacles position. It is clear in the upper view that some UAVs execute evasions of the fixed obstacles (for instance A_3 and A_6) and they also avoid each other (for instance A_1 and A_2). The profiles of the control inputs (velocities) of each agent are shown in Fig. 5. We can see that the evasion actions yield changes on the velocities evolution, for instance, UAVs A_3 and A_6 avoid a fixed obstacle around the time $0.5s$ and UAVs A_1 and A_2 avoid each other around the $1s$ time. Notice that the profiles of velocities are continuous all the time thanks to the use of the smooth transition function $h(t)$.

4.2. Predefined-time convergence

This subsection is dedicated to present the performance of the scheme detailed in Theorem 3.3 using the predefined-time consensus control law (33) with $k_f = 20$ and the TBG described by the function $h_p(t) = 2(t/t_f)^3 - 3(t/t_f)^2 + 1$, which fulfills the conditions of Theorem 3.2, i.e., $h_p(0) = 1, h_p(t_f) = 0$ and $\dot{h}_p(0) = \dot{h}_p(t_f) = 0$. The predefined time to reach the formation was set to $t_f = 4s$. The results using the Gazebo simulator are presented in Figs. 6-8. Fig. 6 shows the consensus for the trajectories of the virtual agents. Fig. 6a shows the 3D trajectories of all the virtual agents. It can be seen that the trajectories converge to a common value q_v^* and therefore, the UAVs reach the desired formation according to the displacement vectors v_i. The other subfigures show the evolution of the virtual agents position in each coordinate with

respect to time. The final value of each subfigure is the corresponding consensus value $q^*_{v_x} = 3.79$ (Fig. 6b), $q^*_{v_y} = 2.10$ (Fig. 6c) and $q^*_{v_z} = 2.20$ (Fig. 6d). The consensus value in each coordinate is reached at the predefined time $t_f = 4s$, then the UAVs achieve the specified formation in that time. In this case of predefined-time convergence, we can set a feasible desired time constrained by the physical limitations of the UAVs and that convergence time will be accomplished independently of the initial UAVs positions.



(a) Trajectory of the virtual agents

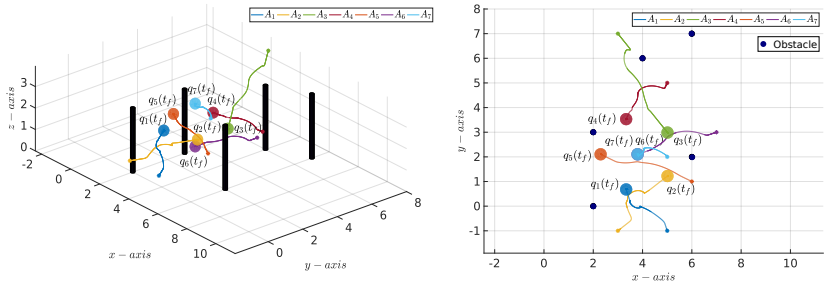
(b) Consensus of agents in x-axis

(c) Consensus of agents in y-axis

(d) Consensus of agents in z-axis

Figure 6: **Predefined-time convergence.** Consensus of virtual agents with convergence in predefined-time of 4s.

The trajectories followed by the UAVs in this experiment are presented in Fig. 7. Fig. 7a shows the 3D trajectories of the UAVs from their arbitrary initial positions to reach the specified formation at time t_f. Fig. 7b shows the upper view of the same trajectories, where it can be observed that some UAVs execute evasion of fixed obstacles (for instance A_3 and A_6) while others also avoid each other (for instance A_1 and A_2). The evolution in time of the UAVs velocities is shown in Fig. 8. It is worth noting that unlike the finite-time control law, the predefined-time approach yields null UAVs velocities at the beginning (time



(a) Isometric view of trajectories

(b) Superior view of trajectories

Figure 7: **Predefined-time convergence.** Trajectories of the UAVs reaching formation and avoiding obstacles.

$t = 0$) and the velocities evolve in a smooth way as the time grows, to finally return to zero at the specified time t_f when the formation is achieved. The generation of smooth control inputs, without discontinuities at the initial time as other control laws, is a good advantage of our predefined-time control law, which is especially beneficial for UAVs. Besides, this behavior of the velocities presented in Fig. 8 is helpful in order to save the energy of the UAVs despite the finite-time convergence. We can also see that the collision avoidance task introduces smooth changes in the velocities profiles.

4.3. Dealing with irregular unknown obstacles

This section is dedicated to show the performance of the scheme detailed in Theorem 3.3 in environments with unknown obstacles with irregular shapes, in particular using the predefined-time control law. The parameters $h_p(t), t_f, k_f$ are taken from the example presented in Section 4.2. The simulation setup corresponds to 2 fixed obstacles and 5 UAVs moving in the x-y plane. Each UAV obtains the relative position of the nearest point from the UAV to the obstacle and when an obstacle is detected in the safety distance the task of obstacle avoidance is activated. Fig. 9 shows the trajectories of each agent, where it can be seen that UAVs avoid the obstacles in the environment and reach the desired formation. Fig. 10 shows the continuous control inputs (velocities) of each UAV. Notice that they initiate in zero and return to zero when the formation

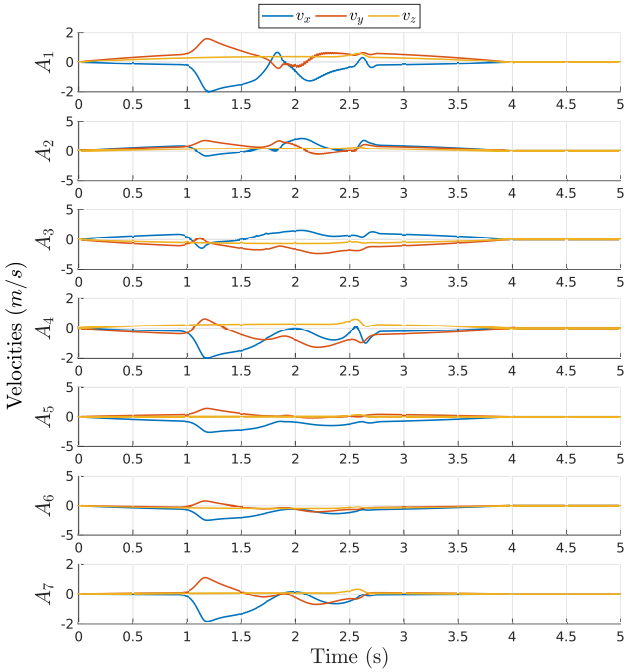


Figure 8: **Predefined-time convergence.** Control inputs (velocities) of the UAVs.

is achieved in 4 seconds. Fig. 11 shows the consensus of virtual agents, where each agent achieves the consensus value at the predefined time.

5. Results of real experiments

In this section, the validation of the control scheme is presented in a real environment with one obstacle and the formation specified for three UAVs Parrot model Bebop 2. These UAVs have a preloaded inner controller that allows us to specify velocity commands. Each UAV is connected via WiFi to a computer where the velocity commands are computed to reach a triangular formation. The position of each UAV and the obstacle in the environment are obtained from an Optitrack Motion Capture System. Due to limitations in the physical space to make the experiments and to have an appropriate safety distance between UAVs, we put one of the quadrotors flying static in hover and the others

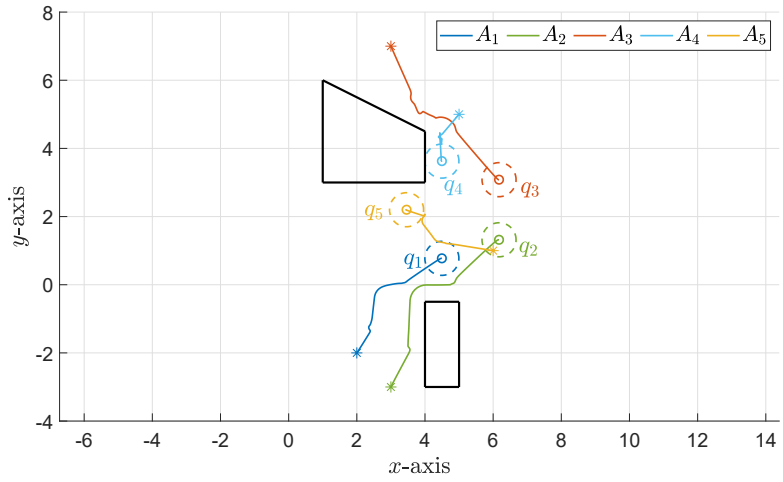


Figure 9: **Predefined-time convergence with irregular obstacles.** Trajectories of each agent.

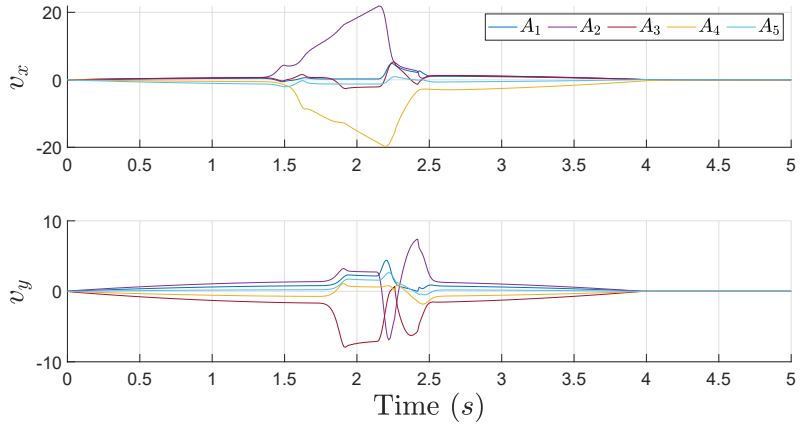


Figure 10: **Predefined-time convergence with irregular obstacles.** Control inputs (velocities) of the UAVs.

have to reach the formation around it autonomously.

5.1. Finite-time convergence

To validate the results of the proposed scheme in Theorem 3.3, experiments were performed using the finite-time convergence control law (25) with $k = 0.02$.

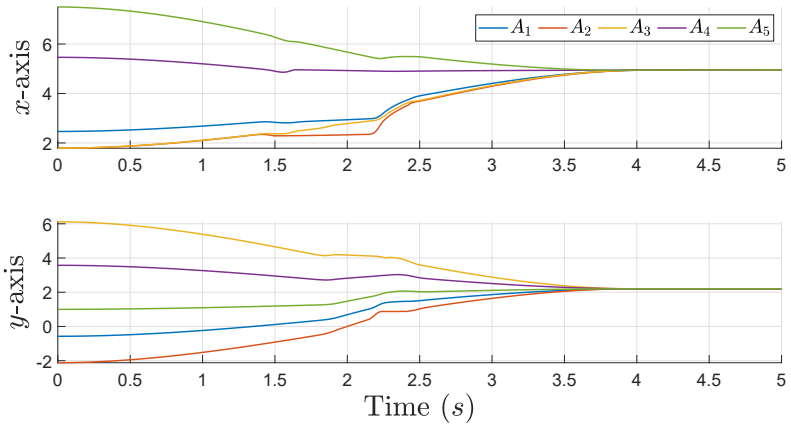


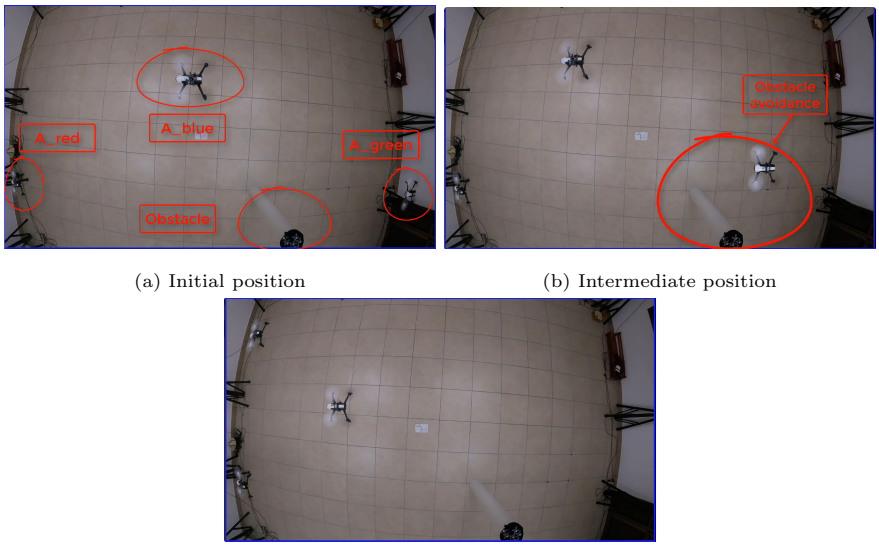
Figure 11: **Predefined-time convergence with irregular obstacles.** Consensus of virtual agents with predefined-time convergence of 4s.

For the experiments with three UAVs Parrot Bebop 2, they are renamed as A_{red}, A_{blue} and A_{green}. In this implementation, the UAV A_{red} flies static and the others must reach the specified triangular formation according to it. Fig. 12 shows snapshots of one of the experiments. In particular, Fig. 12a shows the initial conditions of each UAV and the position of the obstacle. Fig. 12b depicts when the task of obstacle avoidance is activated and finally Fig. 12c shows the final position when the UAVs reach the specified formation.

Fig. 13 shows the trajectories followed by the three UAVs captured by the Optitrack Motion Capture System. It can be seen in Fig. 13a how the UAVs reach the specified formation from arbitrary initial conditions and, at the same time, A_{green} performs an obstacle avoidance action, which is more evident in Fig. 13b. The computed control inputs by the proposed algorithm corresponding to linear velocities of the UAVs are shown in Fig. 14, where around the time $t = 4s$, the task of obstacle avoidance is activated.

5.2. Predefined-time convergence

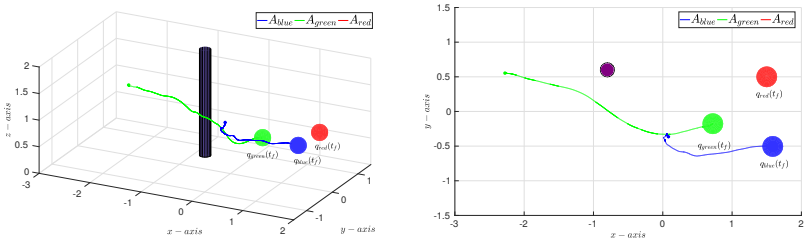
In this subsection, the results of the implementation to validate the performance of the scheme detailed in Theorem 3.3 using the control law (33) for



(a) Initial position (b) Intermediate position

(c) Final position

Figure 12: Snapshots of one experiment with the UAVs Parrot Bebop 2.



(a) Isometric view of trajectories (b) Superior view of trajectories

Figure 13: **Finite-time convergence.** Trajectories of the UAVs Parrot Bebop 2 reaching formation and avoiding obstacles.

predefined-time convergence are presented. The value of the control gain used in the implementation is $k = 0.29$, the TBG used is described by the function $h_p(t) = 2(t/t)^3 - 3(t/t_f)^2 + 1$ and the predefined time is $t_f = 20$. Due to the dynamical features of the quadrotors and their own inner controller, the tracking of the TBG function is not as accurate as in the simulations, which made complicated the realization of experiments with obstacles in the constrained available space. Therefore, we present the results of one experiment of forma-

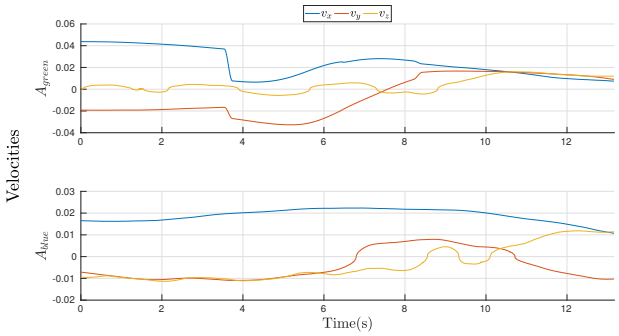
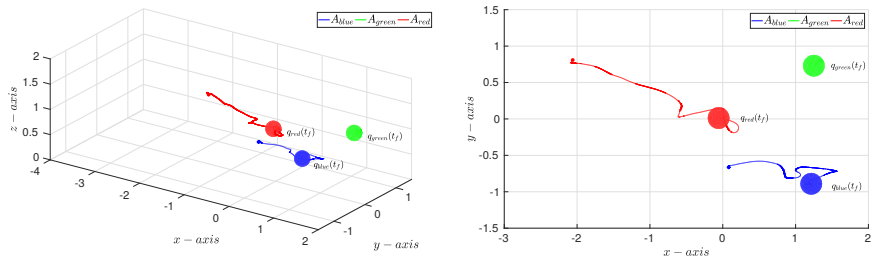


Figure 14: **Finite-time convergence.** Control inputs (velocities) of the UAVs Parrot Bebop 2 .

tion in predefined-time without obstacles in the environment and controlling in the $x - y$ plane.

In this implementation, the UAV named A_{green} flies static and the UAVs A_{red} and A_{blue} must reach the formation in a predefined-time according to the A_{green} position. Fig. 15 presents the trajectories of the UAVS to achieve the specified formation in the predefined time. The isometric view of the trajectories is shown in Fig. 15a while an upper view is shown in Fig. 15b. The computed velocities given by the proposed algorithm for the controlled UAVs A_{red} and A_{blue} are shown in Fig. 16. Notice in this figure that the control inputs present oscillations, which may be due to the own inner controller of the UAVs. We could not modify the time-response of the controller and its accuracy despise small values in the velocity commands. Therefore, we use a stop condition given by a threshold of the consensus error norm, such that this is accomplished close to the desired 20 seconds of convergence time, as shown in Fig. 17. This verifies that the specified triangular formation has been successfully achieved.

A video of the simulation results and the real experiments, showing the performance of the approach during the interaction of both formation and obstacle avoidance tasks, can be found in the following link:
`https://youtu.be/loRUiCldnUo`.



(a) Isometric view of trajectories

(b) Superior view of trajectories

Figure 15: **Predefined-time convergence.** Trajectories of the UAVs Parrot Bebop 2 reaching formation and avoiding obstacles.

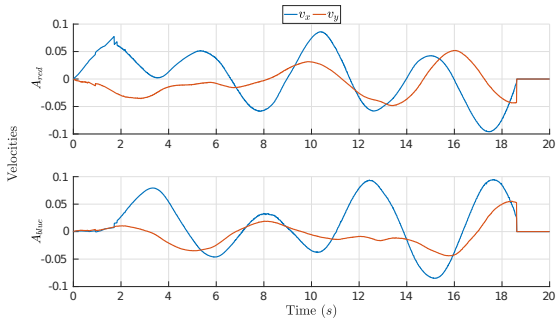


Figure 16: **Predefined-time convergence.** Control inputs (velocities) of the UAVs Parrot Bebop 2 .

6. Conclusions

This paper presented a distributed solution to the problem of position formation of a group of UAVs moving in a 3D environment while avoiding obstacles, using a hierarchical task-based scheme for this purpose. In this case, two tasks are considered, the one with higher priority is devoted to avoid obstacles and the one with lower priority to the UAVs formation. The solution interference between both tasks is avoided in this scheme and a continuous time-dependent transition function between tasks is used to maintain the continuity of the control inputs (UAVs velocities). The formation control has been solved by two novel control laws that can be used independently: the first proposed control law allows the UAVs to achieve a specified formation in finite time, while the

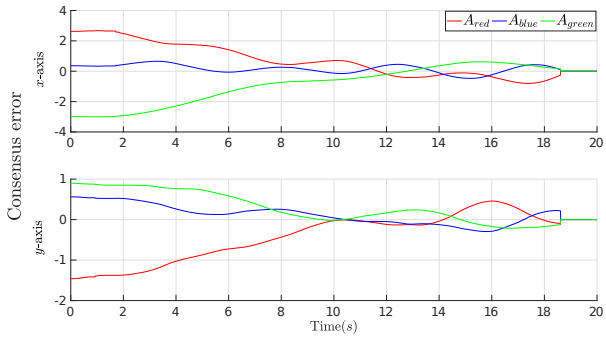


Figure 17: **Predefined-time convergence.** Consensus error in x-axis and y-axis.

second control law achieves a specified formation in a constant predefined time in spite of the initial agents state. The convergence of the proposed solutions are proved theoretically and their effectiveness is shown in simulations and real experiments with a group of quadrotors. As future work, we will address the problem of trajectory tracking in formation, i.e. navigation of the whole UAVs formation by tracking a reference given by a leader and a group of followers able to avoid obstacles of the environment using local information. Besides, we will extend our results for varying topologies, ensuring continuity in the computed velocities. We also plan to address these problems using onboard sensing to directly obtain relative positions, for instance using vision (RGBD cameras) on the UAVs.

References

[1] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proceedings of the IEEE 95 (1) (2007) 215–233.

[2] L. E. Parker, Multiple Mobile Robot Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 921–941.

[3] W. Ren, R. W. Beard, Distributed consensus in multi-vehicle cooperative control, Springer, 2008.

[4] Z. Li, Z. Duan, Cooperative control of multi-agent systems: a consensus region approach, CRC Press, 2014.

[5] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, C. Zhou, Synchronization in complex networks, Physics Reports 469 (3) (2008) 93 – 153.

[6] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, M. Fu, A distributed algorithm for resource allocation over dynamic digraphs, IEEE Transactions on Signal Processing 65 (10) (2017) 2600–2612.

[7] R. Olfati-Saber, R. M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Transactions on Automatic Control 49 (9) (2004) 1520–1533.

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

[9] W. Ren, Multi-vehicle consensus with a time-varying reference state, Systems and Control Letters 56 (7) (2007) 474 – 483.

[10] Y. Shang, Finite-time consensus for multi-agent systems with fixed topologies, Int. J. of Systems Science 43 (3) (2012) 499–506.

[11] Z. Zuo, L. Tie, A new class of finite-time nonlinear consensus protocols for multi-agent systems, International Journal of Control 87 (2) (2014) 363–370.

[12] B. Ning, J. Jin, J. Zheng, Fixed-time consensus for multi-agent systems with discontinuous inherent dynamics over switching topology, International Journal of Systems Science 48 (10) (2017) 2023–2032.

[13] Z. Zuo, Q. L. Han, B. Ning, X. Ge, X. M. Zhang, An overview of recent advances in fixed-time cooperative control of multiagent systems, IEEE Transactions on Industrial Informatics 14 (6) (2018) 2322–2334.

[14] D. Gómez-Gutiérrez, C. R. Vázquez, S. Čelikovský, J. D. Sánchez-Torres, J. Ruiz-León, On finite-time and fixed-time consensus algorithms for dynamic networks switching among disconnected digraphs, International Journal of Control 93 (9) (2020) 2120–2134.

[15] Y. Wang, Y. Song, D. J. Hill, M. Krstic, Prescribed-time consensus and containment control of networked multiagent systems, IEEE Transactions on Cybernetics 49 (4).

[16] J. A. Colunga, C. R. Vázquez, H. M. Becerra, D. Gómez-Gutiérrez, Predefined-time consensus of nonlinear first-order systems using a time base generator, Mathematical Problems in Engineering 2018 (2018) 1957070. doi:10.1155/2018/1957070.

[17] R. Aldana-López, D. Gómez-Gutiérrez, M. Defoort, J. D. Sánchez-Torres, A. J. Muñoz-Vázquez, A class of robust consensus algorithms with predefined-time convergence under switching topologies, arXiv:1812.07545.

[18] Y. Liu, Y. Zhao, W. Ren, G. Chen, Appointed-time consensus: Accurate and practical designs, Automatica 89 (2018) 425 – 429.

[19] D. E. Chang, S. C. Shadden, J. E. Marsden, R. Olfati-Saber, Collision avoidance for multiple agent systems, in: 42nd IEEE Int. Conf. on Decision and Control, Vol. 1, 2003, pp. 539–543.

[20] X. Liu, S. S. Ge, C.-H. Goh, Formation potential field for trajectory tracking control of multi-agents in constrained space, International Journal of Control 90 (10) (2017) 2137–2151.

[21] H. Yu, P. Shi, C.-C. Lim, D. Wang, Formation control for multi-robot systems with collision avoidance, International Journal of Control 92 (10) (2019) 2223–2234.

[22] J. Jin, N. Gans, Collision-free formation and heading consensus of nonholonomic robots as a pose regulation problem, Robotics and Autonomous Systems 95 (2017) 25–36.

[23] O. Cetin, G. Yilmaz, Real-time autonomous uav formation flight with collision and obstacle avoidance in unknown environment, Journal of Intelligent & Robotic Systems 84 (2016) 415–433.

[24] S. Zhou, A. Shen, M. Wang, S. Peng, Z. Liu, Study on composing dense formations in a dynamic environment of multirotor uavs by distributed control, Mathematical Problems in Engineering 2018 (2018) 8.

[25] J. Wu, H. Wang, N. Li, Z. Su, Formation obstacle avoidance: A fluid-based solution, IEEE Systems Journal (2019) 1–12.

[26] E. L. de Angelis, F. Giulietti, G. Rossetti, Multirotor aircraft formation flight control with collision avoidance capability, Aerospace Science and Technology 77 (2018) 733 – 741.

[27] C. Samson, M. L. Borgne, B. Espiau, Robot Control: The Task Function Approach, 1st Edition, Vol. 22 of Oxford Engineering Science Series, Oxford University Press, New York, USA, 1991.

[28] G. Antonelli, F. Arrichiello, S. Chiaverini, Experiments of formation control with multirobot systems using the null-space-based behavioral control, IEEE Transactions on Control Systems Technology 17 (5) (2009) 1173–1182.

[29] G. Arechavaleta, A. Morales-Díaz, H. M. Pérez-Villeda, M. Castelán, Hierarchical task-based control of multirobot systems with terminal attractors, IEEE Transactions on Control Systems Technology 25 (1) (2017) 334–341.

[30] C. Rosales, P. Leica, M. Sarcinelli-Filho, G. Scaglia, R. Carelli, 3D Formation Control of Autonomous Vehicles Based on Null-Space, Journal of Intelligent & Robotic Systems 84 (1-4) (2016) 453–467.

[31] X. Zhu, Y. Liang, M. Yan, A flexible collision avoidance strategy for the formation of multiple unmanned aerial vehicles, IEEE Access 7 (2019) 140743–140754.

[32] C. Godsil, G. Royle, Algebraic Graph Theory, Vol. 8 of Graduate Texts in Mathemathics, Springer-Verlag New York, 2001.

[33] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 2520–2525.

[34] B. Siciliano, Kinematic control of redundant robot manipulators: A tutorial, Journal of Intelligent and Robotic Systems 3 (3) (1990) 201–212.

[35] G. Antonelli, F. Arrichiello, S. Chiaverini, The null-space-based behavioral control for autonomous robotic systems, Intelligent Service Robotics 1 (1) (2008) 27–39.

[36] J. Lee, N. Mansard, J. Park, Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control, IEEE Transactions on Robotics 28 (6) (2012) 1260–1277.

[37] A. Bacciotti, L. Rosier, Liapunov functions and stability in control theory, Springer Science & Business Media, 2006.

[38] S. P. Bhat, D. S. Bernstein, Geometric homogeneity with applications to finite-time stability, Mathematics of Control, Signals and Systems 17 (2) (2005) 101–127.

[39] D. Liberzon, Switching in systems and control, Springer Science & Business Media, 2003.