IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 19, 2020, accepted March 22, 2020, date of publication April 2, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985032

A Quadral-Fuzzy Control Approach to Flight Formation by a Fleet of Unmanned Aerial Vehicles

MARCO ANTONIO SIMOES TEIXEIRA[1], FLAVIO NEVES-JR[1], ANIS KOUBAA[2,3], (Member, IEEE), LUCIA VALERIA RAMOS DE ARRUDA[1], (Member, IEEE), AND ANDRE SCHNEIDER DE OLIVEIRA[1], (Member, IEEE)

[1]Graduate School of Electrical Engineering and Computer Science (CPGEI), Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba 80230-901, Brazil
[2]Prince Sultan University, Robotics and Internet-of-Things Lab (RIOTU), Riyadh 12435 3276, Saudi Arabia
[3]CISTER, INESC-TEC, ISEP, Polytechnic Institute of Porto, 4200-465 Porto, Portuga

Corresponding author: Marco Antonio Simoes Teixeira (marcoteixeira@alunos.utfpr.edu.br)

This work was supported in part by the National Counsel of Technological and Scientific Development of Brazil (CNPq), in part by the Coordination for the Improvement of Higher Level People (CAPES), in part by the Brazilian Ministry of Science, Technology, Innovation and Communication (MCTIC), and in part by the Robotics and Internet-of-Things Lab in Prince Sultan University.

ABSTRACT This paper addresses the control of a fleet of unmanned aerial systems (UAVs), termed as drones, for flight formation problems. Getting drones to fly in formation is a relevant problem to be solved when cooperative cargo transportation is desired. A general approach for this problem considers the coordination of a fleet of UAVs, by fusing all information coming from several individual sensors posed on each UAVs. However, this approach induces a high cost as every UAV should have its advanced perception system. As an alternative, this paper proposes the use of a single perception system by a fleet composed of several elementary drones (workers) with primitive low-cost sensors and a leader drone carrying a 3D perception source. We propose a Quadral-Fuzzy approach to ensure that all drones fly in formation and will not collide with each other or with environment obstacles. We also develop a new way to compute potential fields based on possibility fuzzy (fuzziness) measure with the focus of avoiding collisions between the drones. The proposed approach encompasses four high-coupled intelligent controllers that respectively control the leader and worker drones' motion and implement obstacle and collision avoidance procedures. Simulation results using a fleet of four aerial drones are presented, showing the potential for solving usual problems to flights in formation, such as dodging obstacles, avoiding collisions between the drones, among others.

INDEX TERMS Unmanned aerial vehicles (UAVs), multi-agent systems, distance-based formation, flight-formation control, autonomous flight.

I. INTRODUCTION

Nowadays, unmanned aerial vehicles (UAVs) are used in several applications from military and civilian domains such as forest fire monitoring, surveillance, terrain mapping, and surveying, tracking, disaster management, blood or medical equipment delivery, and others [1]–[5]. Aerial drones are fast, flexible, lightweight, low-cost, and easy to use UAV with the potential to reduce the cost and time in the logistic field. An extensive survey of aerial drones for civilian applications is given in [6]. From this survey, one of the most promising applications of aerial drones is for autonomous cargo transport and delivery by e-commerce retailers and also for express delivery of perishable goods such as food or medicines. However, several challenges must be solved for the service to be effective [6]–[8]: (1) Limited payload: in general, goods must not weigh more than 2 kg; (2) Integration of low-cost sensors and positioning system, that is, several sensors like gyroscope, accelerometer, among others, can be used to create the odometry. The sensor fusion with accurate high location sensors, such as Real Time Kinematic GPS [9], allows to obtain the drone position in a global reference system; (3) Avoid obstacles and collisions: it is necessary to establish a flyable collision-free path in a

The associate editor coordinating the review of this manuscript and approving it for publication was Yongping Pan.

dynamic environment; (4) Communication and connectivity: communication links with the ground control station are need to receive instructions; (5) Landing at specified locations or the use of a parachute to delivery goods, (6) Limited flight range due to energy requirement (battery duration): the traveled distance depends on power transfer efficiency for motor, cruising velocity and power consumption of electronics, (7) Other concerning including government regulation and public acceptation.

This paper addressed challenges 1 to 3 above mentioned. The problem of the limited payload can be circumvented by the use of multiples drones that cooperatively transport the product. However, the use of a fleet of drones introduces new problems such as flight formation, drones communication issues, the need for a distributed system for collision, and obstacles avoidance. In this paper, collision avoidance refers to a drone trying to avoid another drone, while obstacle avoidance means that the drones try to avoid obstacles from the environment.

This work aims to present a novel intelligent and cooperative strategy of load transportation using multiple drones through a *quadral-fuzzy* approach. The added value of this work is the use of a predefined formation for navigation, for a fleet of drones composed of a leader drone equipped with precise 3D perception system and worker drones equipped with low-cost positioning systems [10], [11]. The multiple drones system can autonomously deflect obstacles, thus avoiding collisions and possible damage to load and agents.

Concerning challenge 2, the proposed approach has as the prerogative the use of a unique perception system, with resolution and amplitude to supervise the whole group and environment. The fleet leader computes the goal locations of worker drones based on its perception of the situation at every instant of time, based on the distance to the nearest obstacle. Workers know their locations in the environment using sensor fusion (e.g., accelerometer, gyroscope, and GPS) to estimate the position of the drone in the environment. The goal is to use a single 3D perception sensor in the leader to reduce the cost of the system and avoid overlapping information from multiple 3D sensors.

Based on such sensor data, the fleet leader can compute the flyable collision-free path to the entire formation (challenge 3). Moreover, each worker drone is also equipped with a collision-avoidance system based on potential field [12].

Our main contribution lies in the design and development of a highly-coupled *quadral-fuzzy* approach to managing the multiple-agent motion applied to cooperative transport. A fuzzy controller is developed to control the leader motion, considering the current position of the drone and its desired position, as well as the environment perception information to deviate from obstacles. Worker drones act with a similar fuzzy controller, but without the obstacle deviation skill, i.e., their position is defined by the leader. Another fuzzy system performs obstacle deviation for the whole formation and ensures the optimal configuration for cargo transportation. A fuzzy self-preservation strategy is adopted to prevent collisions, assuring a safe flight to drones and cargo.

This work is divided into five sections. Section 2 brings some related works. Section 3 describes in detail the proposed *quadral-fuzzy* approach. Section 4 discusses the approach evaluation based on some experiments results. At last, section 5 presents the work conclusions.

II. RELATED WORKS

Several studies have contributed to making cargo delivery by drone a reality [6]. The researches are looking for improving navigation capabilities such sensing ability [13], [14], intelligent control [15]–[19] and obstacle and collision avoidance [12], [20], [21].

The integration of visual sensing techniques in drone applications is a trend for researches on position-attitude control, pose estimation and mapping, obstacle detection as well as target tracking [13], [22]. Following this trending, we use a 3D perception source providing a cloud of points (or PointClouds), which can collect spatial information from the environment that is combined with information from low-cost sensors (multi-fusion sensor), allowing run procedures for collision and obstacle avoidance and also for path planning.

Nowadays, fuzzy systems have been successfully used for navigation, guidance, and control of autonomous vehicles and mobile robots [23]–[25]. This extensive use is explained by the simple control structure and also the natural and practical design of fuzzy systems [26]. A survey of nonlinear and adaptive intelligent control techniques for a quadcopter drone, as the drones used in this paper, is given in [15] in which the use of fuzzy control is highlighted. For example, the work in [27] develops an autonomous drone, able to follow planned trajectories by using a robust fuzzy controller based on a precise dynamic and kinematic models of the drone. Different from [27] but similar to some works cited in [15], the proposed *quadral-fuzzy* approach developed herein does not require any model and can adapt to unforeseen situations, providing excellent coverage of wide-ranging operating conditions.

Formation control is an essential issue for the development of collective and collaborative behaviors in multi-agents systems. Potential field and leader-follower are the two main approaches for formation control [12]. Hybrid approaches, combining both theories, are often used to build and move formations because they are effective, robust, and easy to handle [28]–[31]. In this paper, besides the use of fuzzy theory to develop intelligent controllers for drones motion and obstacle avoidance as usual in literature [15], [20], we combine potential field and leader-follower approaches to develop a fuzzy system to avoid the collisions in formation. This fuzzy self-preservation system is based on a fuzzy possibility map in which the potential field reflects the fuzziness of each direction vector in the field. This fuzzy potential field computation is a contribution of this work. Finally, all fuzzy controllers in this paper are modeled as recommended

by [32], using centroid defuzzification, conjunction in minimum, disjunction in maximum, activation in minimum and accumulation in maximum.

It is worthwhile to note that the main contribution of this work is an original solution (*quadral-fuzzy* approach) for the multiple drones flight formation. Cargo transportation has been only cited as a motivating example for flight formation problems.

III. THE PROPOSED *quadral-fuzzy* APPROACH

In this section, we propose a multi-drone formation flight strategy for cooperative cargo transportation. We consider two main safety requirements in the design of the system:

- all drones must maintain formation (position and orientation);
- all drones must avoid collision among them and with the environment's obstacles to assure their integrity.

For this purpose, the proposed approach adopts a leader-workers configuration developed through four subsystems: leader control, worker control, self-preservation, and formation maintenance. In this configuration, the leader drone can perceive the environment and presents an active/reactive behavior while workers exhibit only reactive behavior. Each subsystem is an independent fuzzy system that models the corresponding individual (leader/workers) or collective (team) behavior. All subsystems run in a highly-coupled way composing the new *quadral-fuzzy* approach for cooperative cargo transportation, as shown in Figure 1.

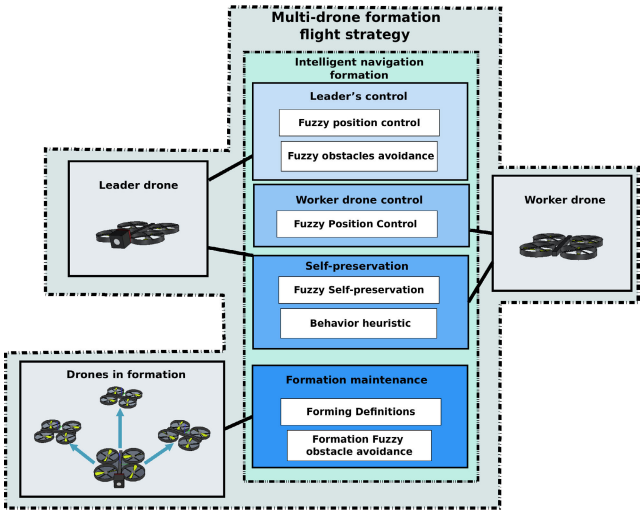

FIGURE 1. schematic of the complete proposed system for multi-drone formation flight (*quadral-Fuzzy* approach).

In this Figure, the first block is related to the *leader control* subsystem that generates the position and orientation control for the leader drone, including obstacle avoidance skill. The obstacle deviation maneuvers should not affect the other drones in the formation; thus, only linear deviations are allowed. The second block refers to the *worker control* subsystem that controls the worker drones' position, assuring a global formation stabilization. Each worker is endowed with the *worker control* subsystem. The third fuzzy system concerns the *self-preservation* ability, where a knowledge-based method is developed to avoid collisions inside formation that can damage the drones. All fleet members are endowed with this security system. Finally, the block called *formation* is responsible for establishing the flying formation rules assuring cargo transportation safety. This module computes the positions of workers around the leader and develops an intelligent strategy for detecting and avoiding external obstacles. This subsystem manages the entire fleet.

In the proposed *quadral-fuzzy* approach, all fuzzy subsystems correspond to fuzzy non-linear functions $g(x)$ that map fuzzy input variables ($x \in U_x \subset \Re^n$) to fuzzy output variables ($y = g(x) \in U_y \subset \Re$). The universe of discourse is defined as $U = [0, lim]$ or $U = [-lim, lim]$, according to the mapped variables. The membership functions describing the fuzzy sets $W_x = \{(x, \mu_A(x)|x \in U_x\}$ and $W_y = \{(y, \mu_A(x)|x \in U_x\}$ are pseudo-trapezoid ones defined in $\Re$, and given by Eq. (1), where $[a, d] \in \Re$; $a \leq b \leq c \leq d$ and $a < d$ ([32]).

$$\mu_A(x; a, b, c, d) = \begin{cases} \dfrac{x-a}{b-a}, & if\ x \in [a, b) \\ 1, & if\ x \in [b, c] \\ \dfrac{x-d}{c-d}, & if\ x \in (c, d] \\ 0, & if\ x \in \Re - (a, d) \end{cases} \quad (1)$$

All fuzzy controllers are based on product inference engine with center average defuzzifier.

The aerial vehicles modeled in this paper is a quadcopter drone with six degrees of freedom corresponding to the linear and angular velocities about X, Y, and Z axis in a 3D environment. Such degrees of freedom are labeled as V_x, V_y and V_z for linear motion and W_x, W_y and W_z for angular movement (roll, pitch, and yaw) as shown in Figure 2.**A**.

The position and orientation of the drone are determined by inertial sensors (i.e., IMU, Gyroscope, GPS). When using multiples drones, all positions must be given in the same coordinate system. Figure 2.*C* shows an example in which the positions of four drones are converted to the same reference frame. Thus, linear and angular transformations based on rotation and translation matrices are used to convert data captured by a sensor to any reference point by the use of transformation trees [33]–[35]. As a result, data from multiples sensors placed on several drones can be translated and processed to the same coordinate system.

In this paper, the coordinates of each drone are transformed into a coordinate in the global frame, that is used as a reference frame for all the drones positioning. Furthermore, homogeneous transformation matrices translate the sensor data to a unique reference point, such as the center of the drone, for example, supporting data processing at the same coordinate system, independent of drone and type of sensor used to data capture. All systems composing the *quadral-fuzzy* approach in Figure 1 are detailed in the next sections.

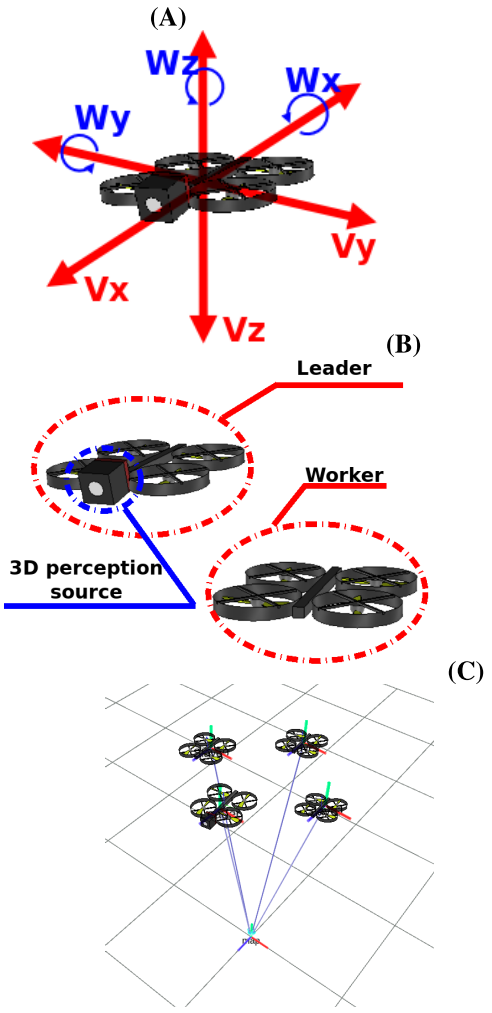

FIGURE 2. Overview of used aerial vehicles (drones). (A) Degrees of freedom of an aerial vehicle (drone). (B) Differences between leader and work drones. (C) Representation of all drones in the same coordinate system.

A. LEADER's CONTROL SYSTEM

A leader drone composes the fleet considered in this paper. This drone can perceive the environment and other drones and also it can estimate their displacement. The leader drone is endowed with a source of 3D perception, and the other drones are considered worker drones, as illustrated in Figure 2.B.

The 3D perception source embedded in the leader drone delivers a cloud of points at a resolution of 176 by 144 points with a viewing angle of 69° (h) x 56°(v). This source can operate up to 10 meters away, and the simulated sensor is based on the Mesa SR4000 3D ToF sensor [36]. This perception source allows detecting the environment's obstacles by the leader drone. All drones (leader and workers) are identical, but only the leader has a source of perception.

The *leader control system* aims to ensure a safe flight for the leader drone. It consists of two fuzzy subsystems, as shown in Figure 3. The *Fuzzy position control* is the motion controller driving the drone to reach the desired position while the *Fuzzy avoid obstacles* block determines leader

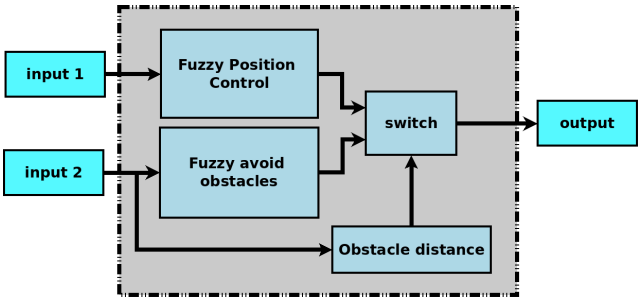

FIGURE 3. Representation of the leader drone position control system. The two Fuzzy controls (Fuzzy position control and Fuzzy avoid obstacles) compete for drone leader speed control, where selection takes place by the distance of identified obstacles.

maneuvers to avoid obstacles while trying to achieve the desired point. Moreover, the block called *switch* turns off the motion controller in the presence of imminent collision, leaving the leader free to carry out obstacle detour, under *Fuzzy avoid obstacles* control.

1) FUZZY POSITION CONTROL SUBSYSTEM

The fuzzy subsystem for position control of the leader drone implements three fuzzy mappings to calculate leader velocities in direction to the desired point that is defined as the goal for the leader, considering the current position of the drone, R_p, and the desired position, D_p. Given in the 3D reference coordinate frame, three measurement distances among these positions are used as input (*input*1 in Figure 3) for this fuzzy subsystem: (1) the Euclidean distance in meters (δ_e) between the position of the drone and the desired point, (2) the angular distance over z-axis that corresponds to the angular orientation error in degrees (δ_a) and (3) linear distance over z-axis computing the linear position error in meters (δ_z). These measured distances are computed by equations 2 and 3.

$$\delta_e = \sqrt{(R_p(x)-D_p(x))^2+(R_p(y)-D_p(y))^2+(R_p(z)-D_p(z))^2} \tag{2}$$

$$\delta_z = (D_p(z) - R_p(z)) \tag{3}$$

The δ_a calculation considers the drone's current position as well as its angular orientation. This input provides the necessary angular rotation so that the drone has pointed to the desired angle. The computational procedure used to obtain δ_a is given by Algorithm 1.

The fuzzy sets $A_i(x)$ are defined for each input variables ($x = [\delta_e, \delta_a, \delta_z]^T$) over their universe of discourse ($\delta_e \in [0, 20]$, $\delta_a \in [-180, 180]$ and $\delta_z \in [-20, 20]$). These fuzzy sets and their correspondent membership functions $\mu_{A_i}(x; a, b, c, d)$ are given in Table 1. Based on these fuzzy sets, the fuzzy values of the euclidean distance δ_e are combined with the other two input variables (δ_a, δ_z) to fire three rule bases generating the output of position controller.

The outputs of the leader's control subsystem are the speeds that will be directly applied to the leader. Although the leader drone is omnidirectional, this control subsystem only

TABLE 1. Input variables of position control: fuzzy sets $A_i(x)$ and membership functions $\mu_{A_i}(x; a, b, c, d)$.

	Euclidean distance (m/s)				Angular distance (°)					Linear z distance (m/s)				
	A_1	A_2	A_3	A_4	A_1	A_2	A_3	A_4	A_5	A_1	A_2	A_3	A_4	A_5
a	0	1	2	3	-180	90	-50	0	50	-20	-4	-2	0	2
b	0	2	3	5	-180	50	0	50	100	-20	-2	0	2	5
c	1	2	3	20	-100	50	0	50	180	-5	-2	0	2	20
d	2	3	5	20	-50	0	50	90	180	-2	0	2	4	20

Algorithm 1 Compute δ_a

```
angleDifference ← antan2(D_p(V_y) − R_p(V_y), D_p(V_x) − R_p(V_x));
if angleDifference < 0 then
    angleDifference ← π + (π − abs(angleDifference));
δ_a ← angleDifference − R_p(W_z);
if abs(δ_a) > π then
    δ_a = 2 * π − abs(δ_a);
    if R_p(W_z) < angleDifference then
        delta_a ← delta_a * −1;
return δ_a;
```

promotes linear motion over $x - z$ plan, preventing sideways sliding. Thus, the three outputs are the linear velocities over the axis x (v_x) and over the axis z (v_z), given in meters per second (m/s), and the angular velocity (ω_z) in radians per second (rad/s). Since decision block (*switch*) in Figure 3, only chooses if the leader's control system output (*output*) comes from *Fuzzy position control* or *Fuzzy avoid obstacles* subsystems, such velocities are also the output variables of position controller. The fuzzy sets $A_i(y)$ defined for each output variables and their correspondent membership functions $\mu_{A_i}(y; a, b, c, d)$ are given in Table 2.

The three non-linear maps (one for each output velocity), generated by the rule bases of fuzzy position control subsystem are shown in Figure 4. As discussed above, the Euclidean distance that corresponds to the 3D position error affects all velocities (angular and linear). It is worthwhile to note that the x-axis linear velocity (v_x) decreases when angular error ($delta_a$) grows-up (see the upper surface in Figure 4). This behavior indicates that the drone must first correct its orientation angle before proceeding to the desired point.

As soon as a goal position is set to the leader drone, the fuzzy position control subsystem continuously computes linear and angular speed that should be applied to the drone's motors according to the decision taken by *switch* block. This decision takes into account the proximity of obstacles, as will be explained in the next section.

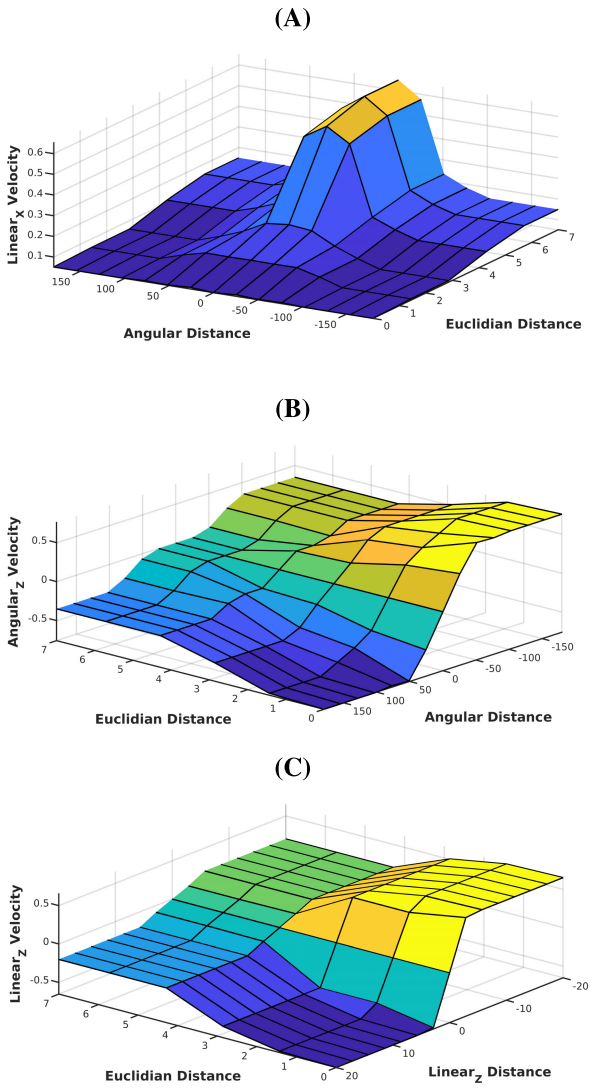

FIGURE 4. Lead drone Position Control - Velocity maps generated by fuzzy position control system. (A) Map of the linear velocity in the X-axis, about the linear error (Euclidean distance) and angular error (angular difference) between the drone position and the desired position. (B) Map of the angular velocity in the Z-axis, about the linear error (Euclidean distance) and angular error (angular difference) between the drone position and the desired position. (C) Map of the linear velocity in the Z-axis, about the linear error (Euclidean distance) and linear Z-axis error (height difference) between the drone position and the desired position.

2) FUZZY OBSTACLES AVOIDANCE SUBSYSTEM

This subsystem aims to prevent collisions with external obstacles. In this paper, any object that is not identified by the leader as a worker drone is considered as obstacles and the formation must deviate it. For this, the cloud of points obtained by 3D perception sensor is processed to detect objects in front of the leader drone, as proposed in [14]. Thus, the closest object is identified, and three distance measurements, in meters, among this object and the leader are

TABLE 2. Output velocities of position control: fuzzy sets $A_i(y)$ and membership functions $\mu_{A_i}(y; a, b, c, d)$.

	$v_x \in [0, 1]$				$\omega_z \in [-1, 1]$					$v_z \in [-1, 1]$				
	A_1	A_2	A_3	A_4	A_1	A_2	A_3	A_4	A_5	A_1	A_2	A_3	A_4	A_5
a	-0.15	0.05	0.15	0.25	-1.5	-0.625	-0.375	0	0.375	-1.5	-0.4	-0.2	0	0.2
b	-0.05	0.15	0.25	0.355	-1	-0.375	0	0.375	0.625	-1	-0.2	0	0.2	0.4
c	0.05	0.15	0.25	1	-0.75	-0.375	0	0.375	1	-0.35	-0.2	0	0.2	1
d	0.15	0.25	0.35	1.5	-0.375	0	0.375	0.625	1.5	-0.2	0	0.2	0.4	1.5

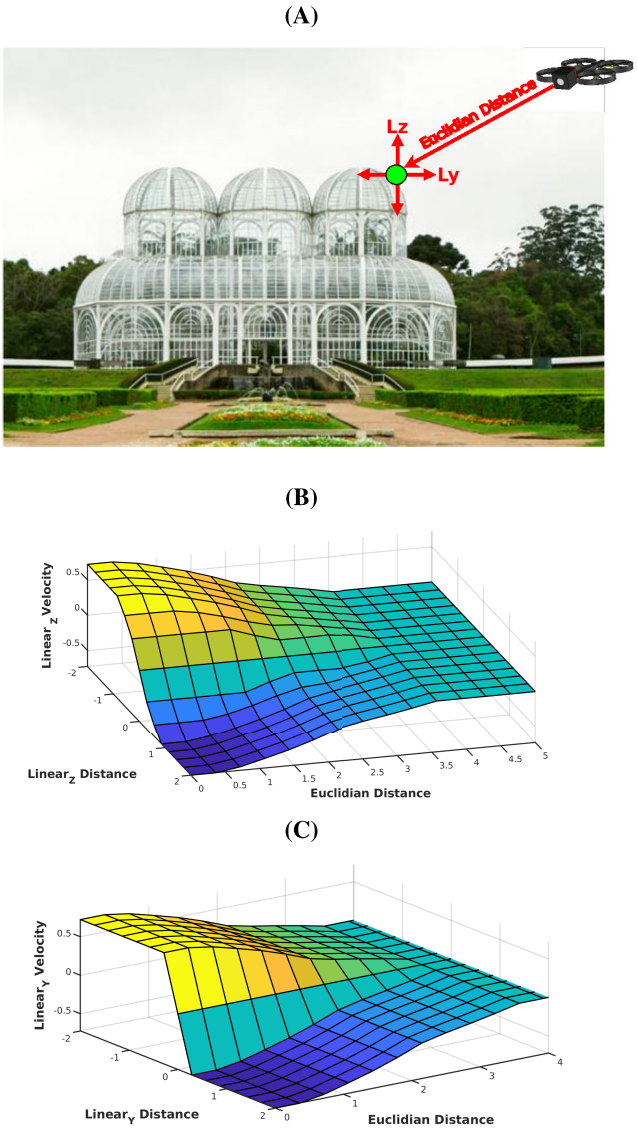

FIGURE 5. Lead drone avoid obstacles control system - Velocity maps generated by fuzzy position control system. (A) Obstacle detection and measured distances to the drone. (B) Map of the linear velocity in the Z-axis, about the linear error (Euclidean distance) and linear error in Z-axis (Z-axis difference) between the drone position and obstacle detected. (C) Map of the linear velocity in the Y-axis, about the linear error (Euclidean distance) and linear error in Y-axis (Y-axis difference) between the drone position and obstacle detected.

computed and used as input (*input*2 in Figure 3) to this subsystem: Euclidean distance (δ_e), the linear distance over the y-axis (δ_y) and the linear distance over the z-axis (δ_z). Figure 5.A illustrates these measured distances, where the green dot represents the obstacle closest to the drone.

Some linguistic predicates are used to define input fuzzy sets. Concerning *Euclidean Distance*(δ_e), the object can be *close* to, *far* or *so far* from the leader. The object position over z-axis (δ_z) can be *in front of*, *above* or *below* the drone. The object position over y-axis (δ_y) can be *in face*, *to the right* or *to the left* of the leader drone. Therefore, the fuzzy set for *Fuzzy obstacles avoidance* subsystem and their correspondent membership functions $\mu(x; a, b, c, d)$ are given in Table 3.

Similar to the position control subsystem, the outputs of this block are speeds in meters per second (m/s) that will be directly applied to the leader drone. The adopted strategy only implements obstacle detour maneuvers in the z - y plan. Thus the leader drone linearly moves along these two axes. Any angular actions are carried out to deflect obstacles; that is, the leader drone is ever oriented to the goal position.

The fuzzy sets $A_i(y)$ and their correspondent membership functions $\mu_{A_i}(y; a, b, c, d)$ for both output variables (*linear velocities* v_y and v_z) are the same and they are given in Table 3. The fuzzy non-linear function (derived from the rule bases) used to compute both velocities are given in Figure 5. Both functions are smooth surfaces, assuring that the leader drone maneuvers to obstacle avoidance do not cause bumps in the cargo.

As the position controller, the obstacle avoidance subsystem is always active; that is, it is always sending speed information to the leader drone. However, the decision about which velocities signals should act on the motors, whether they are the outputs of the position controller, or they come from the obstacle avoidance subsystem, is taken by the *switch* block in Figure 3. For this decision, a simple threshold test is carried out. If the Euclidean distance among the closest obstacles and the leader drone is less than a threshold, then the leader's control system output (*output* in Figure 3) comes from obstacle avoidance subsystem otherwise they are the velocities computed by fuzzy position control subsystem.

B. WORKER CONTROL SYSTEM

The worker drones have only position sensors. Thus, these drones only know their current position relative to their initial position. Moreover, the only worker drone goal is to maintain the formation, and for this, it has a position controller a little simpler than the one described above for the leader. A worker drone safety flight is assured by the *Self-preservation* system that will be described in next section III-C

The worker drone control system can be seen in Figure 6. Its inputs are the same measured distances (δ_e, δ_a, δ_z) of the leader's position controller. However, in this case, the desired

TABLE 3. Fuzzy variables of *Obstacles avoidance* block.

	Input variables									Output variables				
	$\delta_e \in [0,5]$			$\delta_y \in [-2,2]$			$\delta_z \in [-2,2]$			*Linear velocities*				
	Close	*Far*	*So far*	*Rigth*	*Face*	*Left*	*Below*	*Front*	*Above*	**A1**	**A2**	**A3**	**A4**	**A5**
a	-2.5	-0.2	0	-2.5	-0.8	0	-1.5	0.5	2	-1.5	-0.6	-0.3	0	0.3
b	-2	0	0.2	-2	0	0.8	-1	2	3.5	-1.3	-0.3	0	0.3	0.6
c	-0.2	0	2	-0.8	0	2	0.5	2	5.5	-0.6	-0.3	0	0.3	1.3
d	0	0.2	2.5	0	0.8	2.5	2	3.5	6	-0.3	0	0.3	0.6	1.5

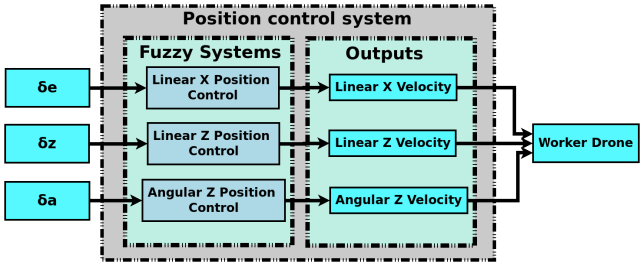

FIGURE 6. Representation of the workers drone position control system.

position corresponds to the position that the worker drone should occupy in the formation. The system outputs also are the same: the linear speeds (v_x and v_z), and the angular velocity (ω_z). However, the position control of the worker drone is organized into three independent fuzzy functions, one for each velocity to be computed. Unlike the leader drone who does not need agility but precision, the workers must take quick actions to keep their place in the formation. Thus, the use of 3 independent controllers provides agility enabling better results in worker drones control.

As the leader and workers are the same kinds of drones, the only difference among them is their embedded perception system, the fuzzy sets and memberships functions for worker drones input/output variables are the same given in Figure 4. However, the universe of discourse for such variables has been expanded ($v_z, \omega_z \in [-4,4]$), narrowed ($\delta_e \in [0,1]$ and $\delta_z \in [-5,5]$) or maintained ($\delta_a \in [-180,180]$ and ($v_x \in [0,1]$) to assure the agility requirements.

The rule base developed to drive the worker drone along x-axis contains rules modeling heuristic knowledge such as

- if the worker drone is very close ($\delta_e \in A1$) to desired position then the speed is very slow ($v_x \in A1$).
- if the worker drone is far ($\delta_e \in A3$) from desired position then speed is fast ($v_x \in A3$).

Similarly, the rules driving movements along z-axis are for example,

- if the vertical distance among the worker drone and the desired position is very down ($\delta_z \in A1$) then the speed is positive and very fast ($v_z \in A5$).
- if the worker drone is in the face of ($\delta_z \in A3$) the desired position, then the speed is near zero ($v_z \in A3$).

As a result of rules firing, if the output of the *Linear Z Position control* subsystem in Figure 6 is negative, the drone descends, and if the linear velocity δ_z is positive, the drone goes up. The *Linear Z Position control* goal is vertically to keep the drone as close as possible to its desired position in the formation.

Furthermore, the *Angular Z Position control* subsystem is always looking to keep the worker drone pointed to its desired position in the formation. For this, it computes drone rotation speed (ω_z) around the z-axis in radians per second (rad/s) according to rules such as

- if the angular error among the worker drone and the desired position is very high and positive ($\delta_z \in A5$) then rotation speed to the right is very fast ($\omega_z \in A1$).
- if the worker drone is aligned to ($\delta_z \in A3$) the desired position, then the rotation speed is near zero ($\omega_z \in A3$).

C. SELF-PRESERVATION SYSTEM

A self-preservation strategy is developed to prevent collisions between the drones during flight. This strategy adopts for each drone, a security cube centered at the drone. If any object (another worker drone) is detected over this cube surface, then the drone must perform a detour maneuver. This maneuver is based on the drone's expectation to move towards the obstacle position $P(x, y, z)$. Thus a possibility map is built describing all directions that a drone can take. A direction is represented by a vector linking the drone center to a position $P(x, y, z)$ over the cube's surface. In addition, a fuzzy system is used to infer the degree of possibility (*fuzziness*) associated to each direction in this map [32], considering drones angular and linear velocities, as well as the Euclidean distance among the drone in the center of the cube and the position $P(x, y, z)$ defining the direction. During flight, positions with angular errors into the interval $[-90°, 90°]$ are in front of the drone, on the contrary these points are behind the drone ($\delta_a \in [-180°, -90°] \cup [90°, 180°]$). In the same way, if the linear drone velocity is positive ($v_x > 0$), the drone is approaching the position; otherwise it is flying away from the position ($v_x < 0$).

The first step to build such maps is to generate a vector field around each drone to assign all possible directions (vectors) for drone displacement, as shown in Figure 7.A. Each vector *Pv[i]*, associated with a direction, is uniquely determined by its position $P(x, y, z)$ and *fuzziness*, corresponding to a degree of possibility of the drone moving in this direction. This degree of fuzziness is computed based on drone position and velocity information by a fuzzy system composed of two modules, as shown in Figure 7.B: linear motion and angular motion.

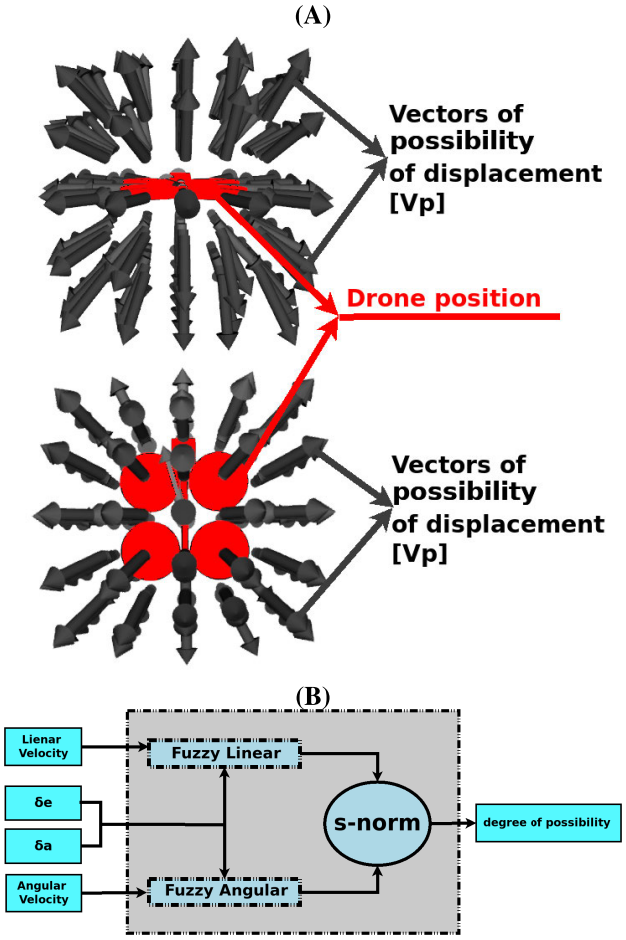

FIGURE 7. Self-preservation system overview. (A) Possibility vector [Vp] scattered around the drone's. (B) Self-preservation system strategy to apply degree of possibility to each possibility vector [Vp].

The first one assigns a fuzziness degree due to the drone's linear velocity uncertainty, weighting forward, or backward drone movements (upper vector field in Figure 7.A). The second fuzzy system considers the angular velocity uncertainty, and it weights turn left or right actions (lower vector field in Figure 7.A). The output of both systems is the degree of possibility of the drone moving in the direction of the vector *Pv[i]*. The s-norm algebraic sum is used to combine these two fuzzy variables giving the final possibility degree of each vector *Pv[i]* [32].

The input variables for both fuzzy systems are the Euclidean (δ_e from equation 2) and angular (δ_a from algorithm 1) distances among the drone and the desired position $P(x, y, z)$ of a *Pv[i]* vector. A third input variable for the fuzzy linear motion system is the linear velocity ($v_x \in [-1, 1]$) over x axis that indicates forward or backward motion. The angular velocity ($\omega_z \in [-1, 1]$) that indicates the left or right rotation is the third input for the fuzzy angular motion system. The input membership sets are the same given above for these variables: euclidean distance and linear velocity are given in Figure 5.D, angular distance, and angular velocity are given in Figure 4.

The output of both fuzzy systems reflects the possibility for the drone to carry out the linear and angular motion in each direction *Pv[i]*. This degree of possibility can be very low (*VL*), low (*L*), high (*H*) or very high (*VH*), in a range from 0 to 1.

For the sake of clarity, the fuzzy surface of both systems is displayed in two graphs combining inputs two by two. For fuzzy linear motion, the rule base has generated the two graphs in Figure 8. From these graphs, when the drone is moving forward, for example, and there are points in the face of it having a low angular difference, that is a small angular error, there is an excellent possibility to the drone reach these points. On the other hand, the fuzzy angular motion system is developed to predict the future position of the drones when performing an angular rotation. Thus *Pv[i]* with a high left angular error, for example, has a higher degree of possibility if the angular velocity is high on the left. The two surfaces in Figure 8 model such inference resulting from fuzzy angular motion rule base.

An example of the two vector fields resulting from both fuzzy linear and angular motion inferences is shown in Figure 9. In this Figure, a high degree of possible results in a light color vector, otherwise a dark vector indicates a direction with a low degree of possibility. The combination (by s-norm algebraic sum) of both fields in Figure 9 generates the final vector field, reflecting the possibilities of the drone moving in each direction.

Finally, a heuristic procedure is developed to prevent collisions based on the vector field computed by a fuzzy motion system. The cube volume around the drone is divided into four quadrants, from 0 to 90 degrees, from 90° to 180°, from −180° to −90° and from −90° to 0°. When another drone is detected in one of these areas, a diversion maneuver is carried out, causing the drone to slide to the opposite side, according to a repulsive force. The degree of probability of the drone moving in a particular direction is used to assign the force of the deviation to be executed in case of another drone has been detected in such direction. All directions *Pv's* that meet the surface of the detected (blue) drone, as shown in Figure 9.A, are identified, defining an interception area. The degree of possibility for all vectors in the interception area is averaged, generating the repulsive force. This deviation maneuver uses the linear velocities on the x and y-axis, weighted by the repulsive force, to move the drone into directions of 45, 135, −135, and −45 degrees from the area containing the detected drone.

When there are two drones around the drone performing the detour maneuver, it calculates the weights (repulsive force) based on the directions *Pv's* connecting it with both detected drones, and then the drones detour is carried out relative to the detected drone with the highest possibility of collision (highest repulsive force).

D. FORMATION MAINTENANCE SYSTEM

This section develops the technique of flying in fleet formation proposed by this work. Firstly, the leader drone is set

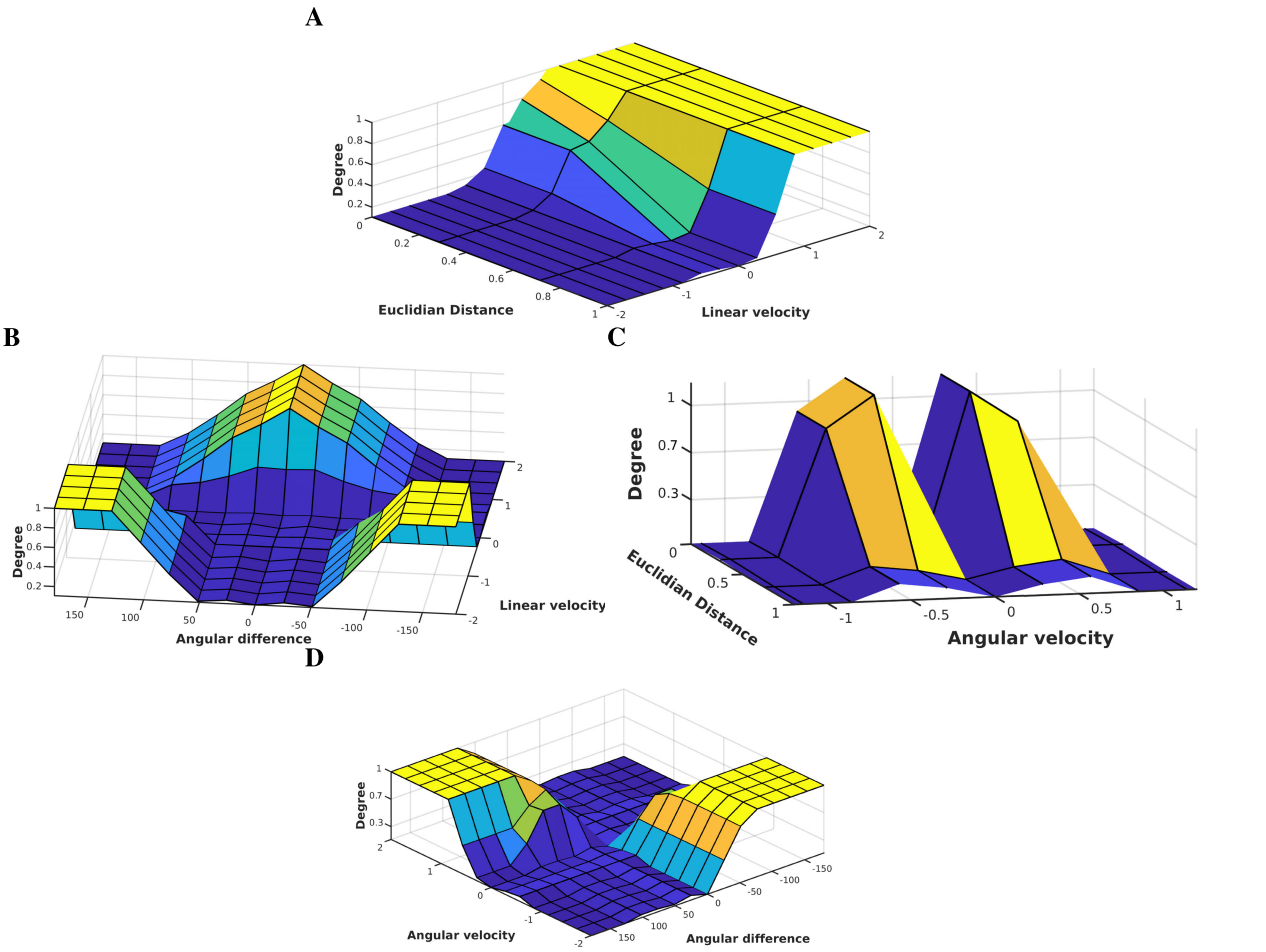

FIGURE 8. Self-preservation system - Degree of possibility maps generated by the Fuzzy linear system [(A), (B)] and Fuzzy angular system [(C), (D)]. (A) Map of degree of possibility, about the linear error (Euclidean distance) and linear velocity (in m/s) between the drones positions. (B)Map of degree of possibility, about the angular error and linear velocity (in m/s) between the drones positions. (C) Map of degree of possibility, about the linear error (Euclidean distance) and angular velocity between the drones positions. (D) Map of degree of possibility, about the angular error and angular velocity between the drones positions.

to reach point B from point A. During the flight, all worker drones must quickly reach specified positions, set from a leader's position, so that it is possible to assume and maintain the formation. The worker drone positions are computed by a formation function specifying the desired geometry shape of the formation. The formation function inputs are the leader position and information about obstacles that have been detected by the 3D perception sensor. There are two types of flight formation: cruiser formation that corresponds to an arbitrary geometrical shape defined according to cargo transportation requirements and safe formation (line mode) to be assumed in the presence of two or more external obstacles.

If the perception sensor identifies only one obstacle in front of the leader drone, the obstacle data is captured (δ_x, δ_z and δ_a) and inputted to a deviation obstacles fuzzy system, computing the necessary sliding maneuver to be carried out by the worker drones around the leader. Thus the formation function computes the new position of each drone in the fleet, assuring that all drones can deflect the obstacle while maintaining the formation. If two or more obstacles are detected in the face or both sides of the leader drone, and the euclidean distances among them are less than 1 meter each, the formation assumes the safe mode. In this safe formation, each drone lines up at a predefined order, forming a shell. As before, the formation function computes the new position of each drone in the fleet.

Both cruiser and safe formations take into account the position of the leader drone to determine the worker's position. During a flight, these relative positions are always the same, concerning the position and orientation of the leader. When the leader drone bypasses a nearby obstacle, for example, the deviation degree computed by the fuzzy system is added to the preset position of each drone, causing it to spin around the leader. Each worker drone tracks its desired position all the time except when it slides to deviation from another drone in the fleet, as explained in section III-C. This proposed strategy for formation maintenance is summarized in Figure 10.

The formation is a unique entity, where each drone is part and plays a role. Virtual markers inform the desired and

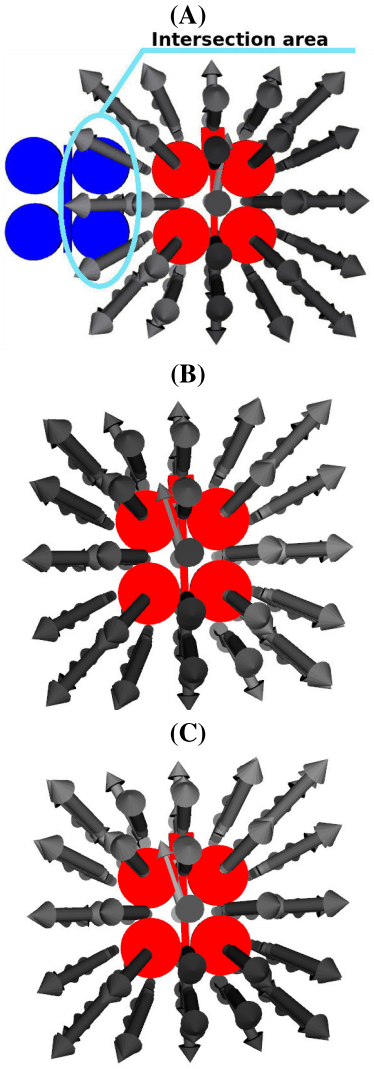

FIGURE 9. Example of self-preservation strategy in operation. (A) Representation of an intersection between two drones. (B) Vectors of possibility indicating that there is a possibility of drone collision in the right side. (C) Vectors of possibility indicating that there is a possibility of drone collision in the front.

actual position of each drone, as well as define the formation limits. These markers are created from the actual position of each drone, obtained through the position sensors and transformation trees. The Figure 11.A shows such markers.

Considering a 3D coordinate [x, y, z] system, the current position of the leader is written as $L_p = [x, y, z]'$ and the actual position of the i-th worker drone is $W_p[i] = [x, y, z]'$ (i also refers to the initial position of the drone in formation). The desired position of the working drone to maintain formation is $Wd_p[i] = [x, y, z]'$.

1) CRUISER FORMATION

The first step to establishing the cruiser formation geometrical shape is to choose all drones' relative positions. For this, the initial leader's position $L_p = [x, y, z]'$ is set, and the positions for each worker drone in the formation are

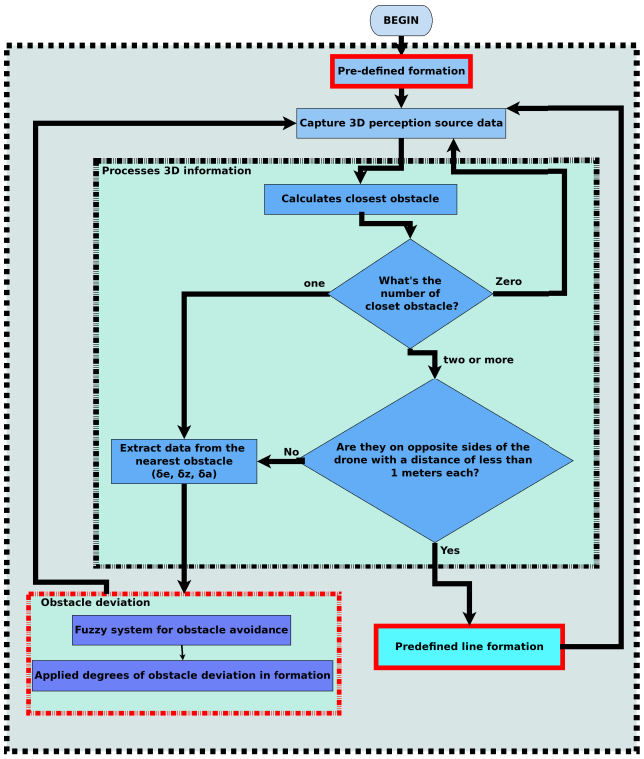

FIGURE 10. Diagram representing all strategy developed for formation maintenance system.The strategy can be divided into 4 main parts, namely *predefined formation, perception source processing, obstacle deviation* and *saved mode* (predefine line formation).

established. Each drone must be at a specific angle and a particular distance from the leader, to assure pose maintenance. A vector $V_d[i]$ indicating the displacement of each drone in each $[x, y, z]$ - axis relative to the leader's position L_p is created to save this geometrical shape. This vector can be added to the leader's current position during the flight, giving the desired new position of each worker drone all the time. However, a $V_d[i]$ has always the same orientation, independent of the leader drone has made an angular displacement. Thus, an angular correction term must be introduced, allowing to compute every time, the worker drone desired position

$$\begin{aligned} Wdp[i].x &= ((Vd[i].x * \cos(Lw.z)) \\ &\quad -(Vd[i].x * \sin(Lo.z))) + Lp.x \\ Wdp[i].y &= ((Vd[i].x * \cos(Lw.z)) \\ &\quad -(Vd[i].x * \sin(Lo.z))) + Lp.x \\ Wdp[i].z &= Lp.z, \end{aligned} \tag{4}$$

where we assumes for simplicity that $L_p.x$ corresponds to the values of x-coordinates of vector $L_p = [x, y, z]'$ and Lw represents the rotation of the drone leader (Figure 2.A). The same is valid for all vectors in the equation 4.

In a fleet with four drones, an initial position of the cruiser formation is represented in Figure 12.A. During the flight, the obstacle deviation maneuvers are carried out to always maintain the displacements ($V_d[i]$) according to this position, revising only its orientation.

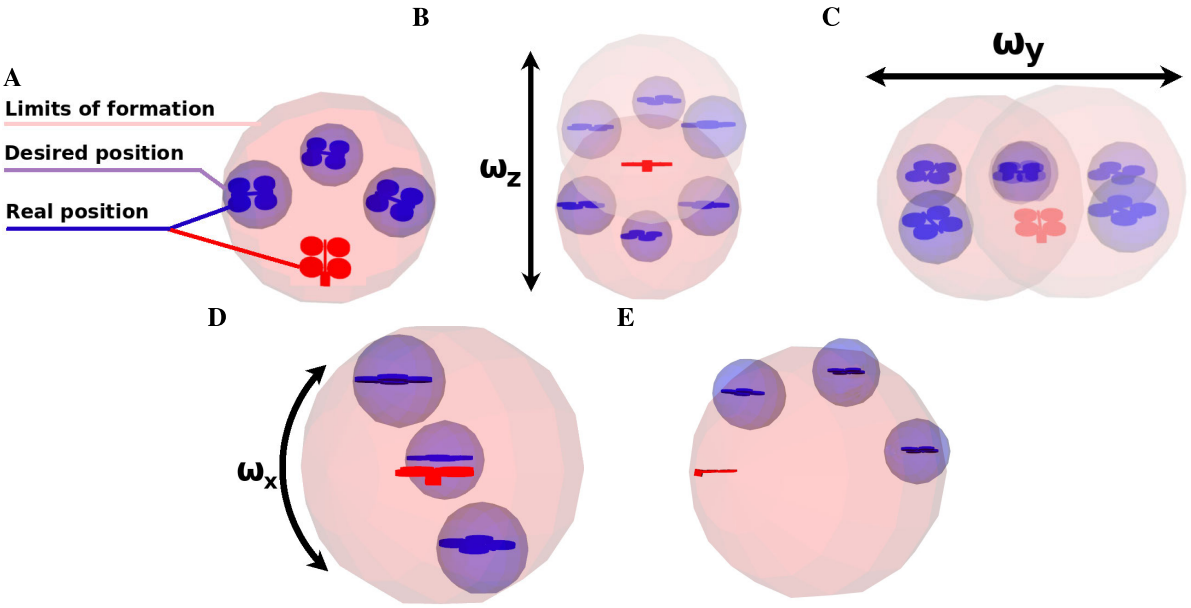

FIGURE 11. The virtual representation of the formation flight, presenting the limits and positions imposed on each drone, as well as the possible deviation maneuvers of the whole formation. (A)Formation limits set by virtual markers, the control proposed by this paper aims to ensure that each drone stays within specific limits. (B) Deviation of all formation in linear Z-axis (when there are obstacles above or below the formation). (C) Deviation of all formation in linear Y-axis (when there are obstacles to the right or left of the formation). (D) Deviation of all formation in the angular X-axis. (E)Line formation (when there are obstacles on both sides).

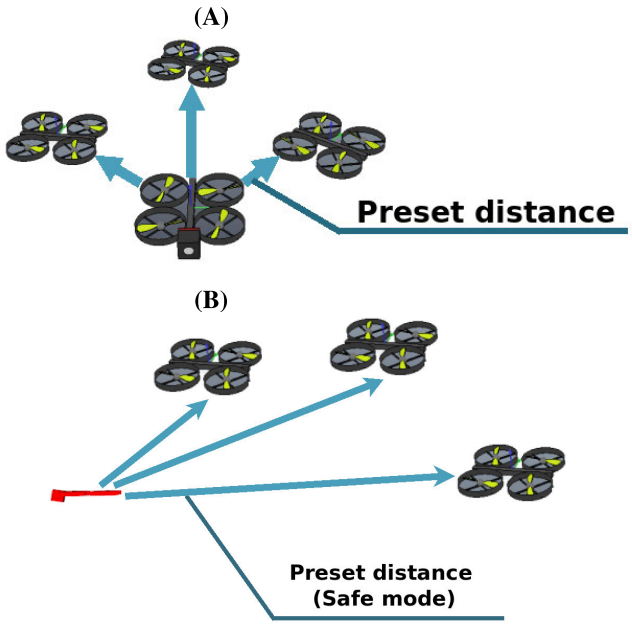

FIGURE 12. Pre-established drone formation. (A) Example of drones' position in cruiser formation. (B) Example of drones' position in safe formation (line mode).

2) SAFE FORMATION

When there are obstacles on both sides of the leader drone, at a distance of fewer than one meter on each leader side, the formation must assume the *safe mode*. It consists of a predefined geometrical shape in which worker drones form a shell behind the leader drone, causing all drones to pass through the same space. This shape assures no drone will hit off the charge, unlike what would happen if the formation rotated 90 degrees along the x-axis (roll rotation V_x in Figure 2.A). The worker drones' relative position computation is also given by Equation 4, considering that the distance vector (V_d) now corresponds to position for this safe formation as presented in Figure 12.B.

3) OBSTACLE DEVIATION

In the presence of an external obstacle, workers can slide around the leader to maintain formation while avoiding the obstacle. Rotation and translation matrices are used to modify the desired position of each worker drone in order to carry out the detour maneuver. The rotation movement changes the position of the worker drone without drawing it from the formation. In this way, the preset distance showed in Figure 12.A is always kept; only the corresponding vector is rotated around the leader drone.

Therefore, the procedure for obstacle deviation by the formation can be implemented into two steps. In the first one, the deviation obstacles fuzzy system infers the angle needed to execute the detour. In the second step, the desired position of each drone is rotated by this angle, considering the leader orientation.

The obstacle information used by the deviation obstacles fuzzy system is the same input variables of the fuzzy obstacles avoidance system in section III-A.2. They are δ_e, δ_y and δ_z. Their universes of discourse, fuzzy sets, and associated membership functions are the same given in Figure 5.D. δ_e refers to the Euclidean distance in meters between the leader and

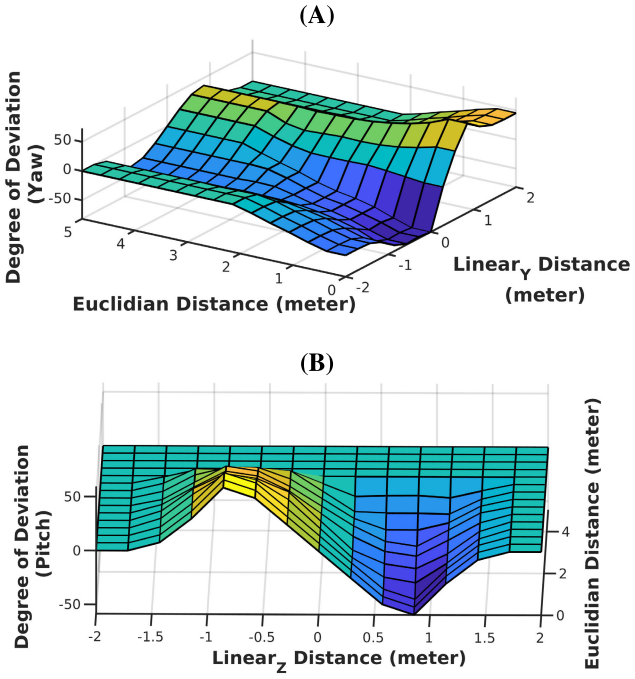

FIGURE 13. Surface graph of the Fuzzy system used for the diversion of obstacles in formation. (A) Degrees applied to the deviation in Yaw, about the linear error (Euclidean distance) and linear error in Y-axis (Y-axis difference) between the leader drone position and obstacle detected. (B) Degrees applied to the deviation in Pitch, about the linear error (Euclidean distance) and linear error in Z-axis (Z-axis difference) between the leader drone position and obstacle detected.

the closest detected obstacle. δ_y refers to the orientation of the obstacle, considering the leader drone current position, whether it is on the right or the left of the leader, and how many meters. δz refers to the obstacle orientation related to the leader drone, whether it is above or below the leader drone, in meters.

The proposed fuzzy system to define the angular deviation of the formation has two outputs: the yaw angle (δ_{yaw}) around the Z-axis and the pitch angle (δ_{pitch}) around Y-axis (V_z and V_y rotation in Figure 2.A). Both outputs are defined over the same universe of discourse $\delta_{yaw}, \delta_{pitch} \in [-150°, 150°]$ and their fuzzy sets have five triangular membership functions, equally spaced over this universe. Yaw deviation angle (δ_{yaw}) provides an angular value that will be added to the formation calculation so that all the worker drones rotate around the leading drone. If δ_{yaw} is positive, the drones will advance to the right; if it is negative, they will move to the left. In the same way, if δ_{pitch} is positive, the workers move up above the leader's position, if it is negative, the movement is to the bottom.

Figure 13 presents the surface graphs of both fuzzy subsystems. It is possible to observe that closer the obstacle is, the higher the angular deviation. These computed angular deviations are summed up to the initial positions of the cruiser formation discussed in section III-D.1. Then the rotation and translation matrices are applied to compute the news positions to be assumed by all working drones so that the formation always maintains the distance between all the drones, thus preserving the original formation geometrical shape.

Figure 11 illustrates the possible motions for the formation, where ω_z refers to the rotation around the leader drone on the Z-axis. This rotation is given by the outputδ_{yaw} of the fuzzy system added to leader drone actual orientation relative to z-axis $Lw.z$. A similar motion is carried out in the y-axis where the rotation ω_y of the working drones around the leader is established from the output δ_{pitch} of the fuzzy system added to leader drone actual angle $Lw.y$ relative to the y-axis. The rotation ω_x concerns the ability of the working drones to rotate around the leader drone along the x-axis. This rotation, also shown in Figure 11.D, is only applied in safe formation mode and corresponds to the roll angle δ_{roll} added to leader drone actual orientation relative to z-axis $Lw.z$.

The rotation values ω_x, ω_z, and ω_y are applied in sequence, considering the working drones' position Wdp, the leader position Lp and the vector Vd. This last vector indicates the displacement vector of each worker drone about the leader drone characterizing the formation of geometrical shape (cruiser or safe), as discussed above.

Equation 5 computes the 3D coordinates position values $[x, y, z]$ for the worker drones considering a rotation along x-axis ($\omega_x = \delta_{roll} + Lw.x$).

$$\begin{aligned} Wdp[i].x &= (Vd[i].x); \\ Wdp[i].y &= (Vd[i].y * \cos(\delta_{roll} + Lw.x)) \\ &\quad -(Vd[i].z * \sin(\delta_{roll} + Lw.x)); \\ Wdp[i].z &= (Vd[i].y * \sin(\delta_{roll} + Lw.x)) \\ &\quad +(Vd[i].z * \cos(\delta_{roll} + Lw.x)); \end{aligned} \tag{5}$$

After computing the ω_x rotation, it is possible to compute rotation along z-axis ($\omega_z = \delta_{yaw} + Lw.z.$) through equation 6. Note that Vd is no longer used; instead, we use now Wdp computed by equation 5 that corresponds to the new worker drone position already rotated by ω_x.

$$\begin{aligned} Wdp[i].x &= (Wdp[i].x * \cos(\delta_{yaw} + Lw.z)) \\ &\quad -(Wdp[i].z * \sin(\delta_{yaw} + Lw.z)); \\ Wdp[i].y &= Wdp[i].y; \\ Wdp[i].z &= (Wdp[i].x * \sin(\delta_{yaw} + Lw.z)) \\ &\quad +(Wdp[i].z * \cos(\delta_{yaw} + Lw.z)); \end{aligned} \tag{6}$$

Finally, the rotation along y axis can be computed by Equation 7 ($\omega_y = \delta_{pitch} + Lw.y.$), considering Wdp compute by equation 6.

$$\begin{aligned} Wdp[i].x &= (Wdp[i].x * \cos(\delta_{pitch} + Lw.y)) \\ &\quad -(Wdp[i].y * \sin(\delta_{pitch} + Lw.y)); \\ Wdp[i].y &= (Wdp[i].x * \cos(\delta_{pitch} + Lw.y)) \\ &\quad +(Wdp[i].y * \sin(\delta_{pitch} + Lw.y)); \\ Wdp[i].z &= Wdp[i].z; \end{aligned} \tag{7}$$

All rotations, ω_x, ω_z, and ω_y are computed based on the displacement vector of each working drone and angular

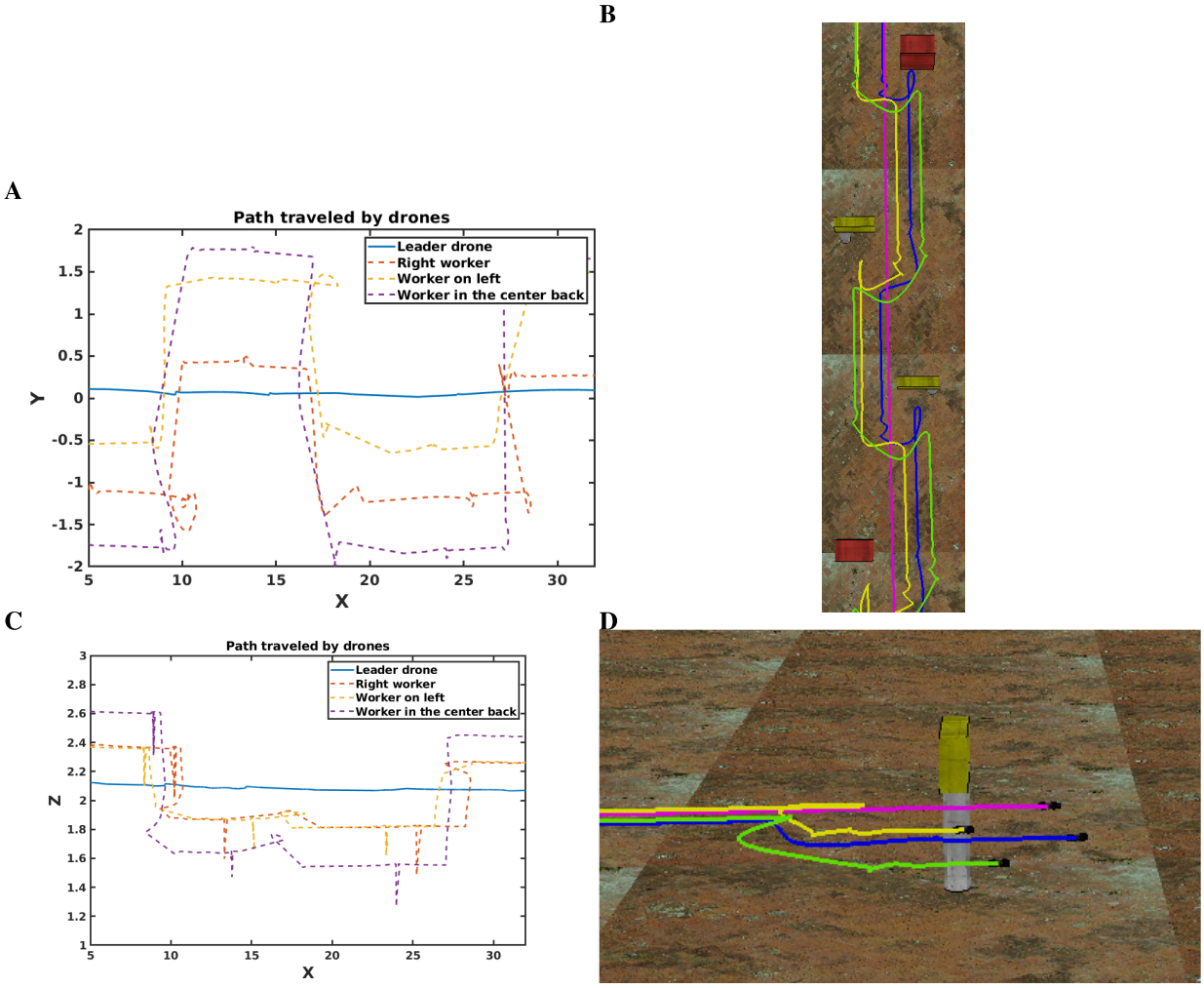

FIGURE 14. Graphical representation of the environment and path taken during navigation. (A) Graph representing the path traveled by all drones on the Y-axis. (B) Graphical representation of the path traveled by the drones on the Y-axis. (V-rep simulator). (C) Graph representing the path traveled by all drones on the Z-axis. (D) Graphical representation of the path traveled by the drones on the Z-axis. (V-rep simulator).

orientation of the leader drone. The resulting position *Wdp* must be now translated, considering the actual position of the leader drone, in order to maintain the formation. This motion is carried out through Equation 8, where *Lp* refers to the actual position of the leader drone in 3D space.

$$\begin{aligned} Wdp[i].x &= Wdp[i].x + Lp.x; \\ Wdp[i].y &= Wdp[i].y + Lp.y; \\ Wdp[i].z &= Wdp[i].z + Lp.z; \end{aligned} \tag{8}$$

However, as discussed in section III-C, the position control of the drone interleaves between avoiding collisions with other drones or going to the desired position. The result of the formation maintenance module only generates a new desired point; the decision about to reach or not the desired position depends on the self-preservation system.

IV. EXPERIMENTAL RESULTS

The proposed *quadral-fuzzy* approach has been tested and validated through simulated experiments using the *Virtual Robot Experimentation Platform* (V-Rep) [37]. Firstly, each module has been one to one tested, and finally, the complete system was validated.

This simulated experiment is carried out with a four drones fleet. The cruiser formation has the leader in front of the platoon and in the back line, a worker drone just behind the leader, one on its left and the other on the right, as shown in Figure 12.A. The three workers drones in the backline keep the same Euclidean distance from the leader.

A scene with several obstacles has been build to asses *quadral-fuzzy* approach performance. This scene is shown in Figure 14.B contains several obstacles that force the fleet to perform deviation maneuvers in the three axes of the 3D space. The drones must traverse the whole scene without colliding with any obstacle and with each other while maintaining the formation.

The position of the four drones is monitored throughout the course. Figure 14) presents the complete traveled paths by all drones during the simulation. The goal is to verify if the formation is maintained during flight navigation, even

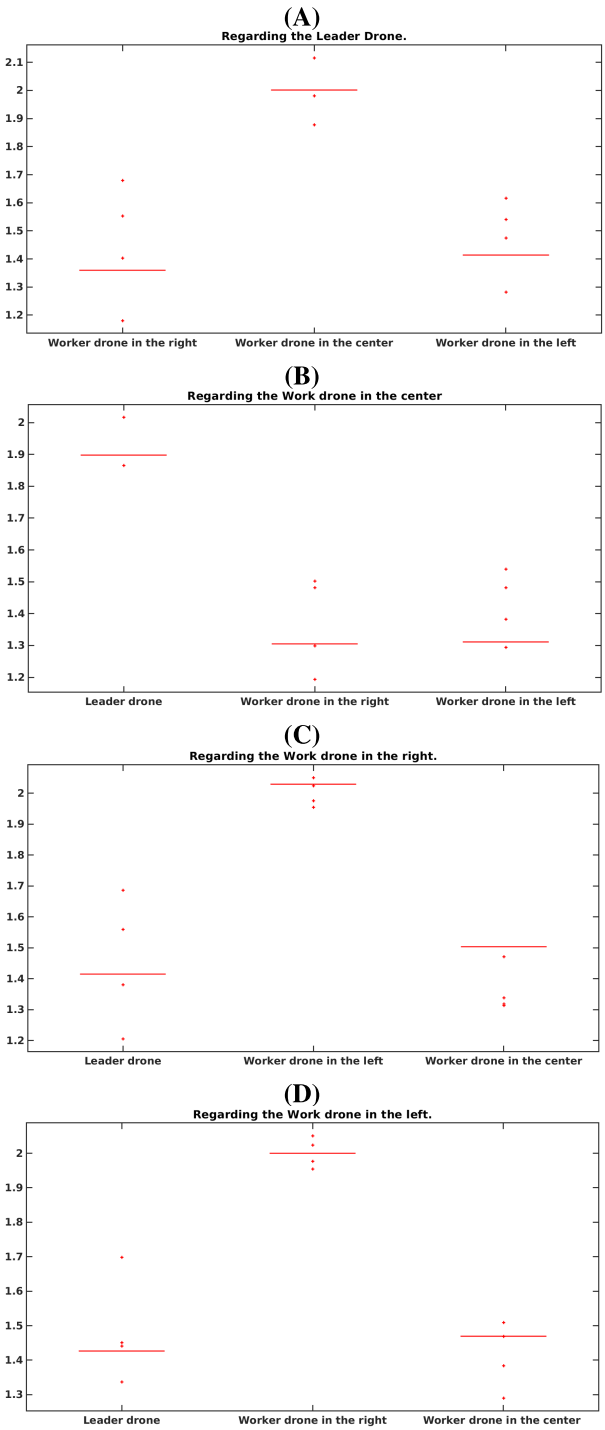

FIGURE 15. Boxplot representing the results obtained during navigation. (A) Distance from working drones to drone leader. (B) Distance between the drones in relation to the middle drone. (C) Distance between the drones in relation to the right drone. (D) Distance between the drones in relation to the left drone.

in the diversion of obstacles. An analysis of the distance between the drones is carried out to show the efficacy of the self-preservation system. Figures 14) A and C show, respectively, the paths developed by all drones in the X-Y and X-Z plans. Figures 14) B and D are respectively, the top views of the trajectories in the X-Y plan and the side view of the trajectories in the X-Z plan. From the latest figures, we can note that some of the drones have sometimes flown in the loop (ellipses shape). This occurred when the leader drone identified an obstacle close to the worker drone that has been forced to carry an evasive action to avoid collisions. After the maneuver, the drone accomplishes a loop to resume its path.

During the simulation, the position of each drone was taken in a rate of 0.2 seconds, when the Euclidean distance between them was calculated. Those measurements are presented in the boxplots at Figure 15 having each drone as reference: Figure 15.A shows the distances from the leader to all other drones, while Figure 15.B shows the Euclidean distance of the middle worker drone to the others, and so on.

As discussed in previous sections, the goal is to keep the distance between the drones throughout the navigation, causing them to fly in formation and do not crash at any moment. As shown in the results, this objective was reached, showing that most of the time the distance between the drones was preserved, with the exception of a few moments, where the obstacle diversion action is performed, moving from a maximum of 20 centimeters.

V. CONCLUSION

This work has presented a *quadral-fuzzy* approach for flight formation using multiple drones. This approach can be applied for several purposes, such as cooperative transport of goods, or patrolling in multi-robot environments, for example. The approach is composed of four central systems: leader drone control, worker drones control, self-preservation, and formation maintenance.

The leader drone commands the platoon since this drone is equipped with a 3D sensor to map the environment. The leader's flight control is toggled between two subsystems: position control and obstacle bypass control. The switching between these two subsystems can determine a change in the positions of worker drones in order to maintain formation. The position control of worker drones is very simple. It only receives a point to be reached and moves the drone towards it. This desired position can be modified at any time during a flight. The self-preservation system prevents collision among the drones inside the formation. The deviation maneuvers are based on the possibility (*fuzziness*) degree of a drone moving in a particular direction. If two drones are detected inside a path intersection area, a repulsive force proportional to their *fuzziness* degree is applied moving away both drones.

The formation maintenance strategy consists in continually assigning positions to the worker's drones based on actual leader drone position and information about external obstacles detected by the perception 3D sensor embedded in the leader drone. In the presence of a detected obstacle, the formation is rotated, causing all drones to deviate from it. If there are several detected obstacles, the fleet can quit cruiser formation, assuming an in-line safety formation until there are no more obstacles. Simulation experiments have been carried out, showing that the proposed approach provides safe

navigation for multiple drones, avoiding collisions between the drones and with obstacles presented in the environment.

REFERENCES

[1] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and YOLOv3," in *Proc. 1st Int. Conf. Unmanned Vehicle Syst.-Oman (UVS)*, Feb. 2019, pp. 1–6.
[2] A. Koubaa and B. Qureshi, "Dronetrack: Cloud-based real-time object tracking using unmanned aerial vehicles over the Internet," *IEEE Access*, vol. 6, pp. 13810–13824, 2018, doi: 10.1109/ACCESS.2018.2811762.
[3] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.
[4] N. Smolyanskiy and M. Gonzalez-Franco, "Stereoscopic first person view system for drone navigation," *Frontiers Robot. AI*, vol. 4, p. 11, Mar. 2017.
[5] D. Tezza and M. Andujar, "The state-of-the-art of human–drone interaction: A survey," *IEEE Access*, vol. 7, pp. 167438–167454, 2019.
[6] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, Dec. 2018.
[7] R. D'Andrea, "Guest editorial can drones deliver?" *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 647–648, Jul. 2014.
[8] E. Tuci, M. H. M. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers Robot. AI*, vol. 5, p. 59, 5 2018. [Online]. Available: https://doi.org/10.3389/frobt.2018.00059
[9] J. Peterson, H. Chaudhry, K. Abdelatty, J. Bird, and K. Kochersberger, "Online aerial terrain mapping for ground robot navigation," *Sensors*, vol. 18, no. 2, p. 630, 2018.
[10] J. Palacin, I. Valgañon, and R. Pernia, "The optical mouse for indoor mobile robot odometry measurement," *Sens. Actuators A, Phys.*, vol. 126, no. 1, pp. 141–147, Jan. 2006.
[11] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, 2004, p. 1.
[12] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of UAV formation flight using 3D potential field," *Simul. Model. Pract. Theory*, vol. 16, p. 1453–1462, Oct. 2008.
[13] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *J. Intell. Robotic Syst.*, vol. 87, no. 1, pp. 141–168, Jul. 2017.
[14] M. A. S. Teixeira, H. B. Santos, A. S. de Oliveira, L. V. Arruda, and F. Neves, "Robots perception through 3D point cloud sensors," in *Robot Operating System (ROS)*, vol. 707, A. Koubaa, Ed. Cham, Switzerland: Springer, 2017, pp. 525–561. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-54927-9_16, doi: 10.1007/978-3-319-54927-9_16.
[15] H. Mo and G. Farid, "Nonlinear and adaptive intelligent control techniques for quadrotor UAV—A survey," *Asian J. Control*, vol. 21, no. 2, pp. 989–1008, Mar. 2019.
[16] M. Varga, J.-C. Zufferey, G. H. M. Heitz, and D. Floreano, "Evaluation of control strategies for fixed-wing drones following slow-moving ground agents," *Robot. Auto. Syst.*, vol. 72, pp. 285–294, Oct. 2015.
[17] J. Hu and A. Lanzon, "An innovative tri-rotor drone and associated distributed aerial drone swarm control," *Robot. Auto. Syst.*, vol. 103, pp. 162–174, May 2018.
[18] C. P. Bechlioulis and K. J. Kyriakopoulos, "Collaborative multi-robot transportation in obstacle-cluttered environments via implicit communication," *Frontiers Robot. AI*, vol. 5, p. 90, Aug. 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2018.00090
[19] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020.
[20] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative UAV formation flying with Obstacle/Collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, pp. 672–679, Jul. 2007.
[21] L. Zhu, X. Cheng, and F.-G. Yuan, "A 3D collision avoidance strategy for UAV with physical constraints," *Measurement*, vol. 77, pp. 40–49, Jan. 2016.
[22] J. Dufek and R. Murphy, "Visual pose estimation of rescue unmanned surface vehicle from unmanned aerial system," *Frontiers Robot. AI*, vol. 6, p. 42, May 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2019.00042
[23] X. Xiang, C. Yu, L. Lapierre, J. Zhang, and Q. Zhang, "Survey on Fuzzy-Logic-Based guidance and control of marine surface vehicles and underwater vehicles," *Int. J. Fuzzy Syst.*, vol. 20, no. 2, pp. 572–586, Feb. 2018.
[24] D. R. Parhi and M. K. Singh, "Navigational strategies of mobile robots: A review," *Int. J. Autom. Control*, vol. 3, nos. 2–3, p. 114–134, 2009.
[25] J. P. L. S. de Almeida, R. T. Nakashima, F. Neves-Jr, and L. V. R. de Arruda, "Bio-inspired on-line path planner for cooperative exploration of unknown environment by a multi-robot system," *Robot. Auto. Syst.*, vol. 112, pp. 32–48, 2019.
[26] J. Alcala-Fdez and J. M. Alonso, "A survey of fuzzy systems software: Taxonomy, current research trends, and prospects," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 40–56, Feb. 2016.
[27] F. Santoso, M. A. Garratt, S. G. Anavatti, and I. Petersen, "Robust hybrid nonlinear control systems for the dynamics of a quadcopter drone," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jun. 4, 2018, doi: 10.1109/TSMC.2018.2836922.
[28] Z. Peng, S. Yang, G. Wen, A. Rahmani, and Y. Yu, "Adaptive distributed formation control for multiple nonholonomic wheeled mobile robots," *Neurocomputing*, vol. 173, pp. 1485–1494, Jan. 2016.
[29] H. Xiao, Z. Li, and C. L. P. Chen, "Formation control of Leader–Follower mobile robots' systems using model predictive control based on neural-dynamic optimization," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5752–5762, Sep. 2016.
[30] X. Zhu, C. Bian, Y. Chen, and S. Chen, "A low latency clustering method for large-scale drone swarms," *IEEE Access*, vol. 7, pp. 186260–186267, 2019.
[31] L. Bian, W. Sun, and T. Sun, "Trajectory following and improved differential evolution solution for rapid forming of UAV formation," *IEEE Access*, vol. 7, pp. 169599–169613, 2019.
[32] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1997.
[33] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control: The Computer Control of Robot Manipulators*. Cambridge, MA, USA: MIT Press, 1981.
[34] W. Khalil and E. Dombre, *Modeling, Identification & Control of Robots*. London, U.K.: Butterworth, 2004.
[35] A. S. de Oliveira, E. R. De Pieri, U. F. Moreno, and D. Martins, "A new approach to singularity-free inverse kinematics using dual-quaternionic error chains in the Davies method," *Robotica*, vol. 34, no. 4, pp. 942–956, Apr. 2016.
[36] MESA IMAGING. (2011). *SR4000/SR4500 User Manual, Version 3.0 Ed.* [Online]. Available: http://downloads.mesa-imaging.ch/dlm.phpfname=customer/Customer_CD/SR4000_SR4500-_Manual.pdf
[37] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

MARCO ANTONIO SIMOES TEIXEIRA received the M.Sc. degree in electrical and computer engineering from the Federal University of Technology—Parana (UTFPR), Brazil, in 2017, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests comprise mobile robots in the areas of perception and intelligent systems, computer vision, navigation, mapping, and applied AI.

FLAVIO NEVES-JR received the B.Sc. and M.Sc. degrees in electrical engineering the from Federal University of Technology—Parana, Brazil, in 1987 and 1989, respectively, and the Ph.D. degree in electrical engineering from University Paul Sabatier (LAAS), France, in 1998. Since 1992, he has been with the Federal University of Technology—Parana, and actually, he is a Full Professor. His research interests comprise hardware and software to instrumentation and automation.

ANIS KOUBAA (Member, IEEE) is currently a Professor of computer science and the Leader of the Robotics and Internet of Things Research Laboratory, Prince Sultan University. He is also a Senior Researcher with CISTER and ISEP-IPP, Porto, Portugal, and a Research and Development Consultant with Gaitech Robotics, China. His current research deals with providing solutions toward the integration of robots and drones into the Internet of Things (IoT) and clouds, in the context of cloud robotics. His research interests also include robot operating systems (ROSs), robotic software engineering, wireless communication for the IoT, real-time communication, safety and security for cloud robotics, intelligent algorithms' design for mobile robots, and multirobot task allocation. He is a Senior Fellow of the Higher Education Academy (HEA), U.K. He has been the Chair of the ACM Chapter in Saudi Arabia, since 2014.

LUCIA VALERIA RAMOS DE ARRUDA (Member, IEEE) received the degree in electrical engineering from the Federal University of Ceara, Brazil, in 1985, the M.Sc. degree from the Graduate School of Electrical Engineering, Campinas State University (FEE/UNICAMP), Brazil, in 1988, and the Ph.D. degree in electrical engineering from the University of Nice-Sophia Antipolis, France, in 1992. Since 1995, she has been with the Federal University of Technology—Parana, and actually, she is a Full Professor. Her research interests comprise soft computing methods to model and control of dynamic systems.

ANDRE SCHNEIDER DE OLIVEIRA (Member, IEEE) received the M.Sc. degree in mechanical engineering, focused in force control of rigid manipulators, from the Federal University of Santa Catarina (UFSC), in 2007, and the Ph.D. degree in engineering of automation and systems, in 2011, with thesis focused on differential kinematics through dualquaternions for vehicle-manipulator systems. He is currently an Adjunct Professor with the Federal University of Technology—Parana (UTFPR). He is a member of the Advanced Laboratory of Robotics and Embedded Systems (LASER) and the Automation and Advanced Control System Laboratory (LASCA). His research interests include the areas of robotics, mechatronics, and automation with special focus in navigation and localization of mobile robots; autonomous and intelligent systems; perception and environment identification; cognition and deliberative decisions; and human-interaction and navigation control.