

APPLIED RESEARCH

Cooperative Threat Engagement Using Drone Swarms

JORGE A. RICARDO JR.¹, LUIZ GIACOMOSSI JR.², JOÃO F. S. TRENTIN^{1,3},
JOSÉ F. B. BRANCALION³, MARCOS R. O. A. MAXIMO², AND DAVI A. SANTOS¹

¹Mechanical Engineering Division, Mechatronics Department, Aeronautics Institute of Technology (ITA), São José dos Campos, São Paulo 12228-900, Brazil

²Autonomous Computational Systems Laboratory (LAB-SCA), Computer Science Division, Aeronautics Institute of Technology (ITA), São José dos Campos, São Paulo 12228-900, Brazil

³EMBRAER S. A., São José dos Campos, São Paulo 12227-901, Brazil

Corresponding author: José F. B. Brancalion (jose.brancalion@embraer.com.br)

This work was supported in part by EMBRAER S. A., Coordination of Superior Level Staff Improvement (CAPES) financial code 001; in part by the Funding Authority for Studies and Projects (FINEP) under Grant 01.22.0069.00; and in part by the São Paulo Research Foundation (FAPESP) under Grant 2019/05334-0 and Grant 2020/12314-3. The work of Davi A. Santos was supported in part by FAPESP under Grant 2019/05334-0 and Grant 2020/12314-3, and in part by the National Council for Scientific and Technological Development (CNPq) under Grant 304300/2021-7.

ABSTRACT The ability of multiple manned and unmanned aircraft systems to cooperatively engage and disable an aerial threat plays a decisive role in modern warfare scenarios. In this paper, we apply key methods to enable the so-called cooperative threat engagement capability among multiple networked agents, *e.g.*, a swarm of drones, with combat and communication capabilities. In particular, this research combines AI-based decision-making and control techniques for a swarm of loyal wingman drones to coordinate efficient defense actions in a cooperative and autonomous manner. We apply these concepts in a defense scenario that is modeled to analyze the loyal wingman concept, which we consider an interesting testbed for cooperative decision-making and low-level control techniques. The investigated methods were implemented in a realistic 3D UAV simulator for demonstration and evaluation.

INDEX TERMS Cooperative engagement capability, loyal wingman UAV, decision making, manned-unmanned teaming, sliding mode control.

I. INTRODUCTION

The cooperative engagement capability (CEC) is an emerging systems-of-systems capability in which multiple systems coordinate their actions through a network to improve their abilities to effectively perform a given task [1]. The term was coined in the 90s in the Johns Hopkins University Applied Physics Laboratory in an effort to enhance the United States Navy's defense capability. The project aimed to improve the situational awareness of a fleet by sharing tactical information among data link participants and speeding up the decision process regarding the interception, engaging, and neutralization of a flying threat, *e.g.*, a cruising missile or aircraft. With the recent advances in aerial robotics, the CEC concept can be extended to unmanned flying platforms

The associate editor coordinating the review of this manuscript and approving it for publication was Zhong Wu¹.

to promote group behavior towards the achievement of a common purpose via a shared communication network. In this context, drone swarms are of particular interest since they enable the use of several low-cost and even disposable platforms in modern defense or surveillance applications.

Recently, the CEC concept has been studied in modern defensive scenarios using autonomous robot swarms [2], [3], [4], [5], [6]. In particular, [2] studied the two-dimensional decision-making problem using adversarial UAV swarms. The mission of each swarm is to defend its base and destroy the adversarial one. To this end, each UAV is equipped with a weapon. Reference [3] studied the intelligent decision-making for an UAV swarm whose mission is to identify and attack specific targets. Similarly, [5] studied a reconnaissance and attack scenario using heterogeneous UAV swarms. These swarms are divided by subsets containing reconnaissance UAVs, attack UAVs, and

reconnaissance-attack UAVs. On the other hand, [4] addresses the defense problem of a critical area using a swarm of UAVs against a larger one. Reference [6] focused on artificial intelligence models for offensive air-to-air combat scenarios using UAVs equipped with machine guns.

In particular, manned-unmanned teaming (MUM-T) is an essential component of the future unmanned aircraft systems' operational environment. It extends the CEC concept by promoting the seamless integration of unmanned platforms, endowed with some level of autonomy, with manned ones. The unmanned platforms aim to perform given tasks under the command of the manned ones. This integration combines the inherent strengths of both platforms to achieve synergy not seen in single-agent systems, thus enabling a higher degree of effectiveness for the team of networked agents. Specifically, the MUM-T context includes the concept of the loyal wingman (LW), which is an unmanned aerial vehicle (UAV) under the tactical command of a high-level manned or remotely controlled leader [7]. Recently, we have witnessed a rapid emergence of multirotor aerial vehicles (MAVs) [8] in either research and industrial applications where aircraft size, weight, and cost play a critical role in their operation, effectiveness, or feasibility [9]. Therefore, we argue that low-cost experimental MAVs are a suitable alternative to investigate the loyal wingman concept in MUM-T scenarios.

This paper investigates the use of LW MAVs with combat capabilities to cooperatively engage and disable aerial explosive threats. Specially, we highlight that this work is an extended version of our previous work [10], which evaluated a 3D defense scenario where a MUM-T composed of loyal wingmen protects a leader UAV and a critical infrastructure from aerial threats. Here, we present a more detailed investigation of the same scenario. In this paper, the vehicles are modeled as fully actuated MAVs, which can perform independent attitude and position tracking [11], [12]. To tackle the design problems that arise from such context, we proposed a problem breakdown that focuses on both the high-level (HL) decision-making tasks and the low-level (LL) control aspects to enable the CEC among the leader and the loyal wingmen in a defense scenario. The LL control addresses the well-recognized robustness requirement for MAV flight control systems against disturbances and uncertainties by designing a joint geometric attitude-position control law using a multi-input first-order sliding mode strategy. Moreover, to provide the MAVs with collision avoidance capability, the reciprocal acceleration velocity obstacles (AVO) [13] method is used. On the other hand, the autonomous behavior of the MAVs is designed using state-of-art artificial intelligence (AI) techniques, by applying finite-state machines (FSMs) and behavior trees (BTs) decision-making algorithms [14] to select and execute modular behaviors for the LW and the aerial threats. We also focus on how the agents must act cohesively as a team, so that cooperation between the MUM-T can emerge, for this we use the classic Kuhn-Munkres task allocation algorithm [15] to distribute tasks between the MUM-T.

To the best of our knowledge, the main contributions of this paper are 1) an extension of the defense scenario proposed in [10] considering a three-dimensional environment with realistic flight dynamics, 2) the application of the reciprocal AVO collision-avoidance method and a multi-input sliding mode flight control law, and 3) the modeling of AI-based decision-making with FSMs and BTs for the proposed scenario.

The remaining of this paper is organized as follows. Subsection I-A presents the notation used along the paper. Section II presents the scenario of interest. Section III defines the LL and HL problem breakdown. Section IV presents the methodology. Section V shows the interface between the HL commands and the LL. Section VI shows the computational architecture used to simulate the scenario. Section VII evaluates the proposed solution using computer simulations. Finally, Section VIII concludes the paper and shares ideas for future work.

A. NOTATION

Matrices and algebraic vectors are denoted, respectively, by uppercase and lowercase boldface letters, while geometric (Euclidian) vectors are denoted as \vec{a} . Sets are denoted by uppercase calligraphic letters as \mathcal{D} . A Cartesian coordinate system (CCS) is represented as $\mathcal{S}_b \triangleq \{B; \vec{x}_b, \vec{y}_b, \vec{z}_b\}$, with B denoting its origin, and \vec{x}_b, \vec{y}_b , and \vec{z}_b representing the unit geometric vectors along its orthogonal axes. The algebraic vectors corresponding to the projection of an arbitrary physical vector \vec{a} onto \mathcal{S}_b and \mathcal{S}_g are denoted by $a_b \in \mathbb{R}^3$ and $a_g \in \mathbb{R}^3$, respectively. The relation between a_g and a_b is $a_b = D^{b/g} a_g$, where $D^{b/g} \in \text{SO}(3)$ is the attitude matrix of \mathcal{S}_b relative to \mathcal{S}_g and $\text{SO}(3) \triangleq \{D \in \mathbb{R}^{3 \times 3}, |D^T D = I_3\}$ represents the special orthogonal group. The inverse of $D^{b/g}$ is denoted by $D^{g/b}$. The identity matrix of order $n \in \mathbb{Z}_{>0}$ is denoted by I_n . Consider two arbitrary algebraic vectors $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$. The vector inequality $x < y$ means that $x_i < y_i, \forall i \in \{1, \dots, n\}$. The sign of x is defined as $\text{sign}(x) \triangleq (\text{sign}(x_1), \dots, \text{sign}(x_n))$, where

$$\text{sign}(x_i) \triangleq \begin{cases} 1 & x_i > 0 \\ 0 & x_i = 0, \\ -1 & x_i < 0, \quad \forall i = 1, \dots, n. \end{cases}$$

Moreover, the Euclidean norm and component-wise absolute value of x are denoted, respectively, by $\|x\|$ and $|x| \triangleq (|x_1|, \dots, |x_n|) \in \mathbb{R}^n$. Similarly, the component-wise absolute value of a generic matrix $A \in \mathbb{R}^{m \times n}$ is $\text{abs}(A) \triangleq (|A_{i,j}|) \in \mathbb{R}^{m \times n}$, where $A_{i,j}$ is the element of the i th row and j th column of A . The standard basis vectors of \mathbb{R}^3 are denoted by $e_1 \triangleq (1, 0, 0)$, $e_2 \triangleq (0, 1, 0)$, and $e_3 \triangleq (0, 0, 1)$. Let $\vec{a}^{i/g}$ represent an arbitrary physical quantity of \mathcal{S}_i with respect to \mathcal{S}_g ; e.g., along the paper, $\vec{v}^{i/g}$ will denote the velocity of \mathcal{S}_i relative to \mathcal{S}_g . An open sphere of radius $\rho \in \mathbb{R}_{>0}$ centered at $p \in \mathbb{R}^3$, the Minkowski sum of two sets, and the set subtraction are

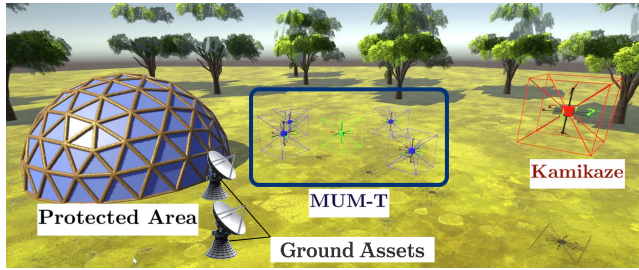


FIGURE 1. Scenario of interest where LW MAVs, highlighted in blue, supported by ground assets, escort a leader MAV and engage kamikaze threats in order to defend a protected area.

denoted, respectively, by

$$\begin{aligned} \mathcal{B}(p, \rho) &= \{x \in \mathbb{R}^3 \mid \|x - p\| < \rho\}, \\ \mathcal{X} \oplus \mathcal{Y} &= \{x + y \mid x \in \mathcal{X}, y \in \mathcal{Y}\}, \\ \mathcal{X} \setminus \mathcal{Y} &= \{x \mid x \in \mathcal{X}, x \notin \mathcal{Y}\}. \end{aligned}$$

Finally, consider the \mathcal{S}_g representations $a_g \triangleq (a_1, a_2, a_3)$ and b_g of \vec{a} and \vec{b} , respectively. The vector product $\vec{c} = \vec{a} \times \vec{b}$ is represented in \mathcal{S}_g by $c_g = [a_g \times] b_g$, where $[a_g \times]$ is the following skew-symmetric matrix:

$$[a_g \times] \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

II. SCENARIO OF INTEREST

To address the concept of CEC among the MUM-T, we define a defense scenario where there are two teams composed of fully actuated MAVs: The MUM-T composed of LW MAVs that fly in formation alongside, *i.e.*, escorting the manned leader, and must defend him as well as a protected area from attacks of an adversarial team, composed by a swarm of kamikaze threats, as depicted in Fig. 1. Therefore, the MUM-T mission is considered to fail if the leader is hit by a single kamikaze, or the protected area is hit by a fixed number of kamikazes. The protected area is equipped with ground assets that provide aerial surveillance capability to the MUM-T members.

The LW is autonomous and capable of making intelligent decisions based on the situational awareness information provided by the ground assets considering that they do not have onboard sensors to obtain information about the threats, such as cameras or LIDAR. To be able to neutralize the threats, the LW is equipped with two hypothetical types of weapons, a mid-range *freezing gun* and a short-range *vaporizer gun*, both with a limited number of cartridges and a fixed cool-down time interval, *i.e.*, they become unavailable for a fixed-time period after shooting. The *vaporizer gun* can neutralize the threat, while the *freezing gun* slows down the threat by half of its maximum speed. The weapons' model is simplified, being the hitting success calculated by a given probability. Note that, the *freezing gun* is intended to make the decision space more complex, in practice, this type of weapon

is infeasible given current technology, and the *vaporizer gun* is also an idealized weapon that uses energy to destroy only the electronic components of the kamikaze.

The leader is remotely controlled by a human and it is in charge of the formation coordination, *i.e.*, it is capable of passing relative coordinate commands to the LW. As a matter of simplification, we assume that the leader can only command the loyal wingmen to fly within one predefined formation pattern. In this paper, we consider this unique pattern as an uniform-circular formation along the local horizontal plane with a desired radius. We also assume that the leader is always capable to command the loyal wingmen whenever required, *i.e.*, it can override the decision-making commands of a specific LW in the MUM-T, either by enforcing the loyal wingmen to return to the formation, or selecting an LW to engage or shoot a specific threat.

On the other hand, the kamikaze MAVs are assumed to explode as soon as they collide with either the leader, an LW, or the protected area. When the explosion is triggered, its effect destroys the target vehicle or damages the protected area. Additionally, once a kamikaze selects a target, it will chase this target until the kamikaze explodes itself or is neutralized by a vaporizer gun. The number of kamikazes is fixed in the simulated scenario, *i.e.*, they immediately respawn after being neutralized, thus keeping a steady stream of attacks. The kamikazes have a simplified AI but are assumed to be faster and more numerous than the MUM-T, thus forcing the MUM-T to cooperatively work in order to effectively neutralize the kamikazes.

The ground assets have a limited detection range and share situational awareness with the MUM-T, *i.e.*, a vector with the state (position, attitude, and linear velocity) of all entities. To simplify the scope of this work, we assume the ground assets to be perfect radars, and that the communication among the MUM-T is perfect, with no delays, packet loss, or bandwidth limitation.

III. PROBLEM BREAKDOWN

In this section, we introduce a problem breakdown according to the scenario of interest. A common approach in robotic cognition involves decomposing the agent into software layers, such as situational awareness, decision-making, and control, as shown in Fig. 2 for a scenario containing a leader and $n \in \mathbb{Z}_{>0}$ autonomous MAVs. In this paper, we use a similar breakdown and focus on both the HL decision-making tasks and the LL control aspects, abstracting the situational awareness block in Fig. 2.

The HL decision-making layer receives state information from the situational awareness containing the pose and linear velocity of each object of interest in a global coordinate system. Once this information is received, the HL layer process the states using AI-based decision-making algorithms that output action commands. There are two types of action commands: rotational and translational movement commands and commands regarding the use of weapons. The LL control, on one hand, translates the movement actions

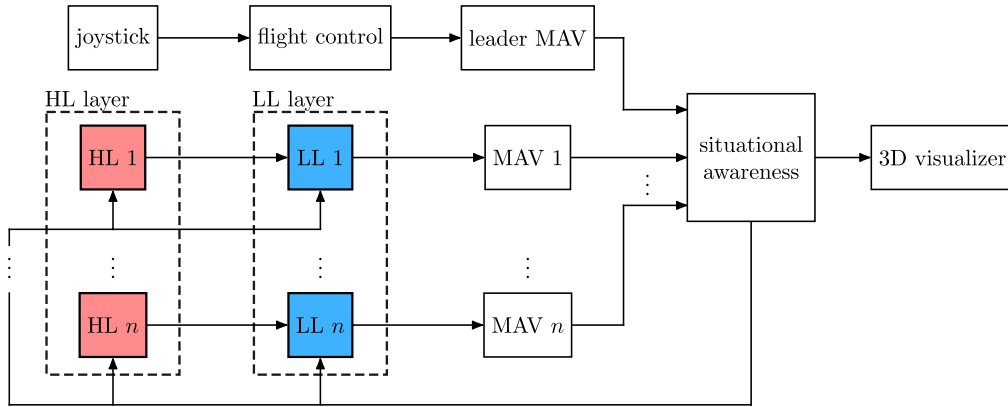


FIGURE 2. Problem breakdown indicating the main software layers.

into appropriate state commands that are used to calculate the control input to the MAV, and, on the other hand, allocates the weapon actions using the available resources. Moreover, the MAV module contains the rotational and translational dynamics of the vehicle, the control allocator that translates the control input into actuator commands, the actuator model, and the weapons model. Note that, the weapons' model is simplified, *i.e.*, the aiming and projectile dynamics are instantaneous, being the hitting success calculated by a configurable probability. Lastly, we also assume that the MAVs share a synchronized internal model of the world in the sense that they are aware of the same threats and other agents state due to the ground-assets radars.

Each agent is identified by a unique ID. In this sense, we define a set $\mathcal{I} \triangleq \mathcal{I}_m \cup \mathcal{I}_k$ containing the identifier IDs of all agents, where \mathcal{I}_m is a set containing the identifier IDs of the MUM-T members and \mathcal{I}_k is a set containing the identifier IDs of the kamikazes.

A. LOW-LEVEL CONTROL PROBLEM

This section presents the translational and rotational dynamics of a general fully actuated MAV equipped with fixed rotors and defines the LL control problem. For a more detailed dynamic modeling, please see [12], [16], [17], and [18].

Consider a ground reference CCS $\mathcal{S}_g \triangleq \{G; \vec{x}_g, \vec{y}_g, \vec{z}_g\}$ located at a known point G on the ground, with \vec{z}_g oriented upwards, parallel to the local vertical. In addition, consider a body-fixed CCS $\mathcal{S}_i \triangleq \{B_i; \vec{x}_i, \vec{y}_i, \vec{z}_i\}$ tied to the airframe of the MAV $i, \forall i \in \mathcal{I}$, with B_i coinciding with its center of mass, as depicted in Fig. 3.

The translational kinematics and dynamics of MAV i are described, respectively, by

$$\dot{r}_g^{i/g} = v_g^{i/g}, \tag{1}$$

$$\dot{v}_g^{i/g} = -g\mathbf{e}_3 + \frac{1}{m_i}f_g^{c,i} + \frac{1}{m_i}f_g^{d,i}, \tag{2}$$

where $r_g^{i/g} \in \mathbb{R}^3, v_g^{i/g} \in \mathbb{R}^3$, and $m_i \in \mathbb{R}_{>0}$ are, respectively, the MAV i position, velocity, and mass, $g \in \mathbb{R}_{>0}$ is the gravity acceleration magnitude, and $f_g^{c,i} \in \mathbb{R}^3$ and $f_g^{d,i} \in \mathbb{R}^3$ are,

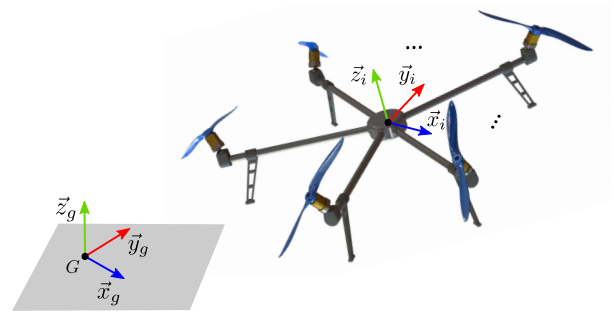


FIGURE 3. Schematic representation of a general fully actuated MAV equipped with fixed rotors along with the adopted CCSs.

respectively, the control and disturbance forces acting on the MAV i .

On the other hand, the rotational kinematics and dynamics of MAV i are represented, respectively, by [19]

$$\dot{D}^{i/g} = -[\omega_i^{i/g} \times] D^{i/g}, \tag{3}$$

$$\dot{\omega}_i^{i/g} = (J_i)^{-1} [J_i \omega_i^{i/g} \times] \omega_i^{i/g} + (J_i)^{-1} (\tau_i^{c,i} + \tau_i^{d,i}), \tag{4}$$

where $D^{i/g} \in SO(3)$ is the attitude, $\omega_i^{i/g} \in \mathbb{R}^3$ is the angular velocity, $J_i \in \mathbb{R}^{3 \times 3}$ is the inertia tensor calculated w.r.t. \mathcal{S}_i , $\tau_i^{c,i} \in \mathbb{R}^3$ is the control torque, and $\tau_i^{d,i} \in \mathbb{R}^3$ is the disturbance torque of the MAV i .

In the proposed scenario, the MAV i moves among static and dynamic obstacles, and is subject to velocity constraint, *i.e.*,

$$r_g^{i/g} \in \mathcal{P}_i(t), \tag{5}$$

$$v_g^{i/g} \in \mathcal{V}_i, \tag{6}$$

where $\mathcal{P}_i(t) \subseteq \mathbb{R}^3$ is the collision-free space, which varies as the obstacles move, and $\mathcal{V}_i \subseteq \mathbb{R}^3$ is an admissible convex set.

We now define the main problem of the LL control.

Problem 1: Execute the HL decisions in the presence of the constraints (5)–(6) and the disturbances $f_g^{d,i}$ and $\tau_i^{d,i}$.

To tackle Problem 1, we adopt the hierarchical control architecture shown in Fig. 4, which contains an outer-loop

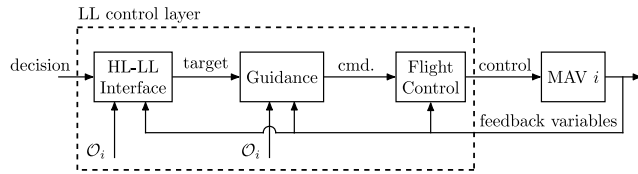


FIGURE 4. Low level control layer of the MAV i . The set \mathcal{O}_i contains information about the other agents' position and velocity.

guidance module and an inner-loop flight control module. Additionally, the high-level to low-level (HL-LL) interface translates the HL decisions into an appropriate target position $\check{r}_g^{i/g} \in \mathbb{R}^3$ and target attitude $\check{D}^{i/g} \in \text{SO}(3)$, which are fed to the guidance law. The HL-LL interface is detailed in Section V. This architecture allows to separate the overall control requirements between the modules. In this sense, the role of the guidance module is to conduct the vehicle to the target pose, while avoiding collisions and ensuring the satisfaction of linear velocity constraints. The control module is entrusted with the task of providing stability and robustness for the motion. We assume that the obstacles' (*i.e.* other MAVs and protected area) position and velocity are available for feedback and are represented by the set \mathcal{O}_i . A similar architecture has been recently investigated in [20] for underactuated MAVs, using robust model predictive control (RMPC) in the guidance loop.

Let us denote the command of a given variable using an overbar symbol, *e.g.*, $\bar{r}_g^{i/g}$ and $\bar{D}^{i/g}$ denote, respectively, the position and attitude commands.

Once the HL decision is translated to a target position and attitude, the guidance law suitably generates the commands $\bar{r}_g^{i/g}$ and $\bar{D}^{i/g}$ to the flight control.

Under the adopted hierarchical control architecture, Problem 1 can be divided into the following two subproblems:

Subproblem 1.1: Design a flight control law to make the MAV i described by (1)–(4) robustly track $\bar{r}_g^{i/g}$ and $\bar{D}^{i/g}$ in the presence of the disturbances $f_g^{d,i}$ and $\tau_i^{d,i}$.

Subproblem 1.2: Design a guidance law to generate $\bar{r}_g^{i/g}$ and $\bar{D}^{i/g}$ to the flight control law to make the MAV i reach $\check{r}_g^{i/g}$ and $\check{D}^{i/g}$ (translated from the HL decision) while respecting constraints (5)–(6).

To tackle Subproblem 1.1, we design a joint attitude-position control law using a first-order sliding mode control (SMC) approach [21]. It uses a high-frequency switching control to drive and keep a given system output to a sliding manifold, where the system ideally becomes insensitive to bounded disturbances of the matched type. Specifically, we design a global first-order sliding mode control (GSMC) that guarantees this insensitivity property during all the time by eliminating the reaching phase [22], [23], [24], under the condition that the control commands are sufficiently smooth. On the other hand, to tackle Subproblem 1.2 and guarantees the smoothness requirement of the control commands, we propose a guidance strategy based on the reciprocal acceleration velocity obstacles (AVO) method [13].

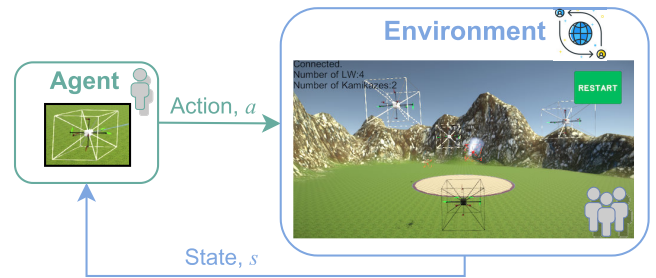


FIGURE 5. Problem 2 definition diagram.

B. HIGH-LEVEL DECISION-MAKING PROBLEM

In this section, the problems to be solved by the decision-making of the LW agents are described.

The main mission of the LW agents is to defend both the protected area and the remotely-controlled leading drone against multiple incursions from a swarm of kamikaze drones. Therefore, an LW must effectively engage and disable threats to ensure the safety of both. Furthermore, for this mission to be successful, LW MAVs are expected to have the ability to make intelligent decisions autonomously. Thus, we can define necessary capabilities for this agent, such as the ability to fly in formation with cohesion, and the ability to engage and neutralize imminent threats to guarantee the defense of both the protected area and the leader. To effectively achieve neutralization, agents must have the capability to use embedded weaponry, *i.e.* use weapons intelligently and strategically, and in extreme cases, sacrifice themselves to neutralize a threat.

The main problem of the HL decision-making can now be defined.

Problem 2: The main problem is to develop an autonomous intelligence module for the LW MAV to successfully achieve the mission objective, aiming at the smallest loss of LW agents during the attack and defense maneuvers.

Based on this problem it is identified a need of defining a set of expected behaviors for the agents. Therefore, we define the following subproblem:

Subproblem 2.1 Design basic behaviors for the LW agents and to develop a decision-making architecture to coordinate these behaviors.

To tackle Subproblem 2.1 it is also necessary to design the adversary agents present in the scenario, so we can properly evaluate our solution to Problem 2.

Subproblem 2.2: Design basic behaviors for the kamikaze agents and to develop a decision-making architecture composed of these behaviors.

Therefore, the main agent in our problem is the LW MAV, who will have to perform actions based on the AI-based decision making, as seen in Fig. 5. The action $a \in \mathcal{A}$, where \mathcal{A} is a set of possible actions, is performed by the agent in a simulated environment which in turn will send the agent's state $s \in \mathcal{S}$ as feedback. Note that the environment is the *scenario of interest* and it encompasses all loyal wingmen agents, the leader, and its kamikaze adversaries.

One of the subproblems to be also addressed is the allocation (or assignment) of tasks among the LW MAVs. By task we mean the neutralization of a specific kamikaze threat.

Despite the LW being autonomous, the agents must act cohesively so that cooperation between the MUM-T can arise. One way to approach this is that the MAV leader must be able to command the LWs when needed, so he must be able to distribute tasks to the MUM-T members. This is known as the classical assignment problem which is a widely-studied problem applicable to many domains [25]. It is the problem of choosing an optimal assignment of $n \in \mathbb{Z}_+$ workers to n tasks, assuming that numerical ratings, or costs, are given for each worker's performance on each task. An optimal assignment is the one that makes the sum of the workers's costs for their assigned tasks a maximum (or minimum) [26]. Note that there are $n!$ possible assignments, of which several may be optimal. In this sense, consider the following subproblem.

Subproblem 2.3: Given a set of workers \mathcal{W} , a set of tasks \mathcal{J} , and a set of costs \mathcal{C} indicating how effectively each worker $w_i \in \mathcal{W}$, where $i \in \{1, \dots, n\}$, can perform each task $j_i \in \mathcal{J}$ determine the best possible assignment of workers to tasks, such that each task is assigned to one worker and each worker is assigned one task, so the total cost is minimized.

One wishes to choose a set of n independent elements $(c_{i,j})$ of a cost matrix C , where $c_{i,j}$ is the element of the i th row and j th column of C , so that the sum of these elements is minimized. This can be expressed as permuting the rows and columns of C to minimize the trace of a matrix $\min_{L,R} Tr(LCR)$, where L and R are permutation matrices, and the cost matrix is

$$C = \begin{matrix} & \begin{matrix} j_1 & j_2 & \dots & j_n \end{matrix} \\ \begin{matrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{matrix} & \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,n} \end{bmatrix} \end{matrix} \quad (7)$$

A well-known solution to the assignment problem is given by the Kuhn-Munkres algorithm, colloquially known as Hungarian algorithm, proposed by Kuhn [15] and refined by Munkres [26]. This algorithm solves the assignment problem assuming the existence of a cost matrix $C \in \mathbb{R}^{n \times n}$, where the element in the i th row and j th column represents the cost $c_{i,j}$ of assigning the j th task to the i th worker.

IV. METHODOLOGY

This section presents the methods used for addressing Problems 1–2. Subsection IV-A describes the method used to solve Problem 1, while Subsection IV-B describes the methods used to solve Problem 2.

A. LOW-LEVEL CONTROL METHODOLOGY

This subsection designs a joint attitude-position control law designed based on a GSMC policy and a guidance strategy based on the reciprocal AVO method for addressing, respectively, Subproblems 1.1 and 1.2.

1) JOINT ATTITUDE-POSITION CONTROLLER

Consider the objective of tracking a time-varying position command $\bar{r}_g^{i/g}$ and attitude command $\bar{D}^{i/g}$ that satisfy the following assumption.

Assumption 1: The time varying commands are such that, at the initial time instant, $\bar{r}_g^{i/g}(0) = r_g^{i/g}(0)$, $\bar{v}_g^{i/g}(0) = v_g^{i/g}(0)$, $\bar{D}^{i/g}(0) = D^{i/g}(0)$, and $\bar{\omega}_i^{i/g}(0) = \omega_i^{i/g}(0)$, and are at least one time differentiable with respect to time, such that their first-time derivatives are Lipschitz continuous.

Assumption 1 is not restrictive; it just requires the knowledge of the MAV i initial position, attitude, and velocities, and the use of suitable differentiable commands.

To design the controller, we derive the equations describing the control errors kinematics and dynamics. To this end, define a command-related CCS $\mathcal{S}_i \triangleq \{\bar{B}_i; \bar{x}_i, \bar{y}_i, \bar{z}_i\}$ representing the commanded position and orientation for the MAV i body-fixed frame \mathcal{S}_i . The attitude and angular velocity control errors can be defined, respectively, as [12], [19], and [27]

$$\tilde{D}^i \triangleq D^{i/i} \equiv D^{i/g} \left(\bar{D}^{i/g} \right)^T \in SO(3), \quad (8)$$

$$\tilde{\omega}^i \triangleq \omega_i^{i/i} \equiv \omega_i^{i/g} - \tilde{D}^i \bar{\omega}_i^{i/g} \in \mathbb{R}^3, \quad (9)$$

where $\bar{D}^{i/g} \triangleq D^{i/g}$ and $\bar{\omega}_i^{i/g} \triangleq \omega_i^{i/g}$ are, respectively, the attitude and angular velocity commands of the MAV i .

Similarly, let us define the position and linear velocity control errors, respectively, as

$$\tilde{r}^i \triangleq \bar{r}_g^{i/g} - r_g^{i/g}, \quad (10)$$

$$\tilde{v}^i \triangleq \bar{v}_g^{i/g} - v_g^{i/g}, \quad (11)$$

where $\bar{r}_g^{i/g} \in \mathbb{R}^3$ and $\bar{v}_g^{i/g} \in \mathbb{R}^3$ are, respectively, the position and velocity commands of the MAV i .

The time derivatives of (8)–(11) can be put, using (1)–(4), into the following state-space model:

$$\dot{x}_1^i = f_1^i \left(x_1^i, x_2^i \right), \quad (12)$$

$$\dot{x}_2^i = f_2^i \left(x_1^i, x_2^i \right) + B^i u^i + B^i d^i, \quad (13)$$

where $x_1^i \triangleq (\tilde{r}^i, \tilde{g}^i)$, with $\tilde{g}^i \in \mathbb{R}^3$ being the Gibbs vector representing $\tilde{D}^{i/g}$, and $x_2^i \triangleq (\tilde{v}^i, \tilde{\omega}^i)$ are the states, $u^i \triangleq (f_g^{c,i}, \tau_i^{c,i})$, $d^i \triangleq (f_g^{d,i}, \tau_i^{d,i})$,

$$\begin{aligned} B^i &\triangleq \begin{bmatrix} -I_3/m_i & 0_{3 \times 3} \\ 0_{3 \times 3} & (J_i)^{-1} \end{bmatrix}, \\ f_1^i \left(x_1^i, x_2^i \right) &\triangleq \begin{bmatrix} \tilde{v}^i \\ \frac{1}{2} \left(\tilde{g}^i (\tilde{g}^i)^T + [\tilde{g}^i \times] + I_3 \right) \tilde{\omega}^i \end{bmatrix}, \\ f_2^i \left(x_1^i, x_2^i \right) &\triangleq \begin{bmatrix} \dot{\tilde{v}}_g^{i/g} + g e_3 \\ (J_i)^{-1} \left[J_i \omega_i^{i/g} \times \right] \omega_i^{i/g} - \tilde{D}^i \dot{\tilde{\omega}}_i^{i/g} + [\tilde{\omega}^i \times] \tilde{D}^i \bar{\omega}_i^{i/g} \end{bmatrix}. \end{aligned} \quad (14)$$

From (13), it can be immediately seen that the term d^i is of the matched type, *i.e.*, it belongs to the span of the input matrix B^i . Consequently, a first-order SMC law is able to guarantee the robustness (or invariance) of the closed-loop system with respect to d^i [28].

Now, let us define the sliding variable

$$s^i \triangleq \dot{x}_1^i + C^i x_1^i, \quad (15)$$

where $C^i \in \mathbb{R}^{3 \times 3}$. The corresponding sliding set is

$$S^i \triangleq \left\{ (x_1^i, \dot{x}_1^i) \in \mathbb{R}^6 \mid s^i = 0 \right\}.$$

From Assumption 1, note that the system is in the sliding set S^i at the initial time instant. Therefore, by designing the control u^i such that the inequality, $\dot{V} \leq -\beta V^{1/2}$ [29], with $\beta \in \mathbb{R}_{>0}$, is satisfied from the initial time instant, it holds that $(x_1^i, \dot{x}_1^i) \in S^i$ during all the time, and consequently $(x_1^i(t), \dot{x}_1^i(t)) = (0_6, 0_6), \forall t \geq 0$. Therefore, from (14), it holds that $x_2^i(t) = 0_6, \forall t \geq 0$. In other words, the system is capable of exactly tracking the position, attitude, linear velocity, and angular velocity commands, *i.e.*, $r_g^{i/g}(t) = \bar{r}_g^{i/g}, v_g^{i/g}(t) = \bar{v}_g^{i/g}, D^{i/g}(t) = \bar{D}^{i/g}(t)$ and $\omega_i^{i/g}(t) = \bar{\omega}_i^{i/g}(t), \forall t \geq 0$.

By differentiating (15) with respect to time and substituting (12)–(13) into the resulting expression, we can obtain the dynamic equation for s^i (we omit the function independent variables for conciseness):

$$\dot{s}^i = C^i f_1^i + \frac{\partial f_1^i}{\partial x_1^i} f_1^i + \frac{\partial f_1^i}{\partial x_2^i} f_2^i + \frac{\partial f_1^i}{\partial x_2^i} B^i u^i + \frac{\partial f_1^i}{\partial x_2^i} B^i d^i. \quad (16)$$

Regarding the disturbance d^i consider the assumption:

Assumption 2: The disturbance d^i is bounded according to $|d^i| \leq d^{i,max}$, where $d^{i,max} \in \mathbb{R}^6$ is a known vector with positive components.

By choosing the control law

$$u^i = -(\Xi^i)^{-1} \left(C^i f_1^i + \frac{\partial f_1^i}{\partial x_1^i} f_1^i + \frac{\partial f_1^i}{\partial x_2^i} f_2^i + K^i \text{sign}(s^i) \right),$$

where $\Xi^i \triangleq \frac{\partial f_1^i}{\partial x_2^i} B^i$ and $K^i \in \mathbb{R}^{6 \times 6}$ is a constant diagonal matrix satisfying

$$K^i 1_6 \geq \text{abs}(\Xi^i) d^{i,max} + 1_6 \frac{\beta}{\sqrt{2}},$$

guarantees the existence of a global sliding mode of s^i in S^i , thus completing the objective of Problem 1. The proof is omitted here but is analogous to the one presented in Section 3.1 of Reference [30].

For implementing the control input u^i , it is necessary a control allocation module, which, for simplicity, has been omitted in Fig. 4. The control allocation generates individual throttle commands to the rotors to make the effective resultant control force and torque close to the respective command. For details about the control allocation of fully actuated MAVs with fixed rotors, one can see reference [12]. In particular, in the simulation study of the present paper, as we have

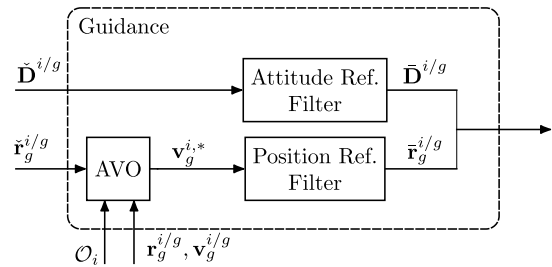


FIGURE 6. Block diagram of the proposed guidance strategy.

considered a non-redundant fully actuated vehicle containing six rotors, the respective control allocation matrix is a square one, and, consequently, the allocation is immediately solved by, first, inverting that matrix and, second, converting the obtained rotor thrusts into throttle commands. For this end, as usual, we assume that the i th rotor thrust is described by $f_i = k_f \omega_i^2$, where k_f is a force coefficient, ω_i is the rotation speed, and $i \in \{1, 2, \dots, n_r\}$, being $n_r \in \mathbb{Z}_{>0}$ the number of rotors.

2) GUIDANCE STRATEGY

This section formulates a guidance strategy based on the reciprocal AVO method to solve Subproblem 1.2.

Let us define a CCS $S_p \triangleq \{P; \bar{x}_p, \bar{y}_p, \bar{z}_p\}$ fixed to a protected area reference point P , with \bar{z}_p oriented upwards, parallel to the local vertical. In the proposed scenario, an LW considers the other MAVs and the protected area as obstacles, while a kamikaze only considers the other kamikazes as obstacles since their main goal is to collide against the MUM-T and the protected area. In this sense, we can define a set \mathcal{I}_i^o containing the ID of all obstacles considered by the MAV i [31].

Now, since the leader is remotely controlled, the guidance strategy presented here for the MAV i is only implemented on the loyal wingmen and the kamikazes, *i.e.*, $i \in \mathcal{I} \setminus \ell$, where ℓ is the leader ID. Moreover, it is important to highlight that, the smoothness requirement of the leader commands $\bar{r}_g^{\ell/g}$ and $\bar{D}^{\ell/g}$ can be achieved by properly generating the leader commands.

In order to guarantee the smoothness requirement of the commands $\bar{r}_g^{i/g}$ and $\bar{D}^{i/g}$ of Assumption 1, we propose a guidance strategy that generates these commands using a position and an attitude reference filters, as depicted in Fig. 6. Then, we use the AVO method to choose a target velocity $v_g^{i,*} \in \mathbb{R}^3$ aiming to avoid collision with the obstacles and respect the velocity constraint (6).

The attitude reference filter is designed using a harmonic-jerk s-curve trajectory generation algorithm [32]. On the other hand, to fulfill the smoothness requirement of the position command, the position reference filter generates the position command for the MAV i by integrating twice the following acceleration command:

$$\ddot{v}_g^{i/g} = \delta_i^{-1} \left(v_g^{i,*} - \bar{v}_g^{i/g} \right), \quad (17)$$

where $\delta_i \in \mathbb{R}_{>0}$.

From (17), it can be seen that by limiting $v_g^{i,*}, \dot{v}_g^{i/g}$ is bounded. Therefore, by choosing $\bar{r}_g^{i/g}(0) = r_g^{i/g}(0)$ and $\bar{v}_g^{i/g}(0) = v_g^{i/g}(0)$, the initial conditions and smoothness requirement of the positing command in Assumption 1 are fulfilled.

Consider that the MAVs and the protected area can be circumscribed by spheres. In this context, a collision might occur between the MAV i , where $i \in \mathcal{I} \setminus \ell$, and Obstacle j , where $j \in \mathcal{I}_i^o$, if

$$\|r_g^{i/g}(t) - r_g^{j/g}(t)\| < \rho_{ij}, \quad (18)$$

where $\rho_{ij} \triangleq \rho_i + \rho_j$, being $\rho_i \in \mathbb{R}_{>0}$ the radius of the sphere enclosing MAV i and $\rho_j \in \mathbb{R}_{>0}$ the radius of the sphere enclosing Obstacle j .

Due to the existence of the global sliding mode in the flight control loop, and considering $v_g^{i,*}$ as a constant input, an expression for the MAV i position can be obtained by integrating (17) two times with respect to time, being given by

$$r_g^{i/g}(t) = r_g^{i/g}(t_0) + \Delta t v_g^{i,*} + g_i(\Delta t) \left(v_g^{i,*} - \bar{v}_g^{i/g}(t_0) \right), \quad (19)$$

where $t_0 \in \mathbb{R}_{>0}$ is an initial time instant, $\Delta t \triangleq t - t_0$, and $g_i(\Delta t) \triangleq \delta_i (e^{-\Delta t/\delta_i} - 1)$.

Similarly to the original VO approach [33], we consider that the Obstacle j keep its velocity constant, *i.e.*, its position can be written as $r_g^{j/g}(t) = r_g^{j/g}(t_0) + \Delta t v_g^{j/g}(t_0)$. Then, substituting (19) and $r_g^{j/g}(t)$ into the collision condition (18) and dividing both sides of the resulting expression by $\Delta t + g_i(\Delta t)$ yields

$$\|v_g^{i,*} - c^i(\Delta t)\| < \frac{\rho_{ij}}{\Delta t + g_i(\Delta t)}, \quad (20)$$

where

$$c^i(\Delta t) \triangleq -\frac{r_g^{i/j}(t_0) - g_i(\Delta t)\bar{v}_g^{i/g}(t_0) - \Delta t v_g^{j/g}(t_0)}{\Delta t + g_i(\Delta t)}. \quad (21)$$

The AVO defines a set of target velocities for the MAV i that will result in a collision with the Obstacle j within a time horizon $\tau \in \mathbb{R}_{>0}$, if the Obstacle j keeps a constant velocity. It can be defined from (20) as

$$\mathcal{AVO}_{i,j}^\tau \triangleq \bigcup_{t_0 < t \leq t_0 + \tau} \mathcal{B} \left(c^i(\Delta t), \frac{\rho_{ij}}{\Delta t + g_i(\Delta t)} \right).$$

The procedure is analogous to the original VO method: continually selecting a velocity outside $\mathcal{AVO}_{i,j}^\tau$ guarantees that the MAV i and Obstacle j will not collide. By using this procedure considering all the obstacles, we guarantee collision-free guidance for the MAV i and consequently satisfy the position constraint (5).

The above procedure is suitably for passive obstacles, *i.e.*, obstacles that do not change their motion based on their surroundings. However, if the moving obstacles are entities capable of making decisions, then this reactive nature has to be considered. This problem is referred in the literature as

reciprocal collision avoidance [34] and can be incorporate into the presented methodology by modifying the AVO set relative to the non-passive obstacles (see Reference [13]).

In the proposed scenario, an LW considers the other LW as non-passive obstacles and the kamikazes as passive ones. On the other hand, a kamikaze considers the others kamikazes as non-passive obstacles.

To account for the velocity constraint (6), it has to be explicitly rewritten in terms of the target velocity $v_g^{i,*}$. To this end, note that, from (17), by choosing $v_g^{i,*} \in \mathcal{V}_i$, it holds that $\bar{v}_g^{i/g} \in \mathcal{V}_i$ since \mathcal{V}_i is a convex set. Then, due to the existing of the global sliding mode, one can show that by choosing

$$v_g^{i,*} \in \mathcal{V}_i,$$

we guarantee that $v_g^{i/g} \in \mathcal{V}_i$. As a result, continuously choosing

$$v_g^{i,*} \in \mathcal{V}_R \triangleq \mathcal{V}_i \setminus \mathcal{AVO}_i^\tau,$$

being $\mathcal{AVO}_i^\tau \triangleq \bigcup \mathcal{AVO}_{i,j}^\tau, \forall j \in \mathcal{I}_i^o$, guarantees the satisfaction of the position and velocity constraints (5)–(6).

B. HIGH-LEVEL DECISION-MAKING METHODOLOGY

In this section, we discuss the methodology used to address Problem 2. For that, we focus on the artificial intelligence of both types of agents present in the scenario of interest. The development of the decision-making architectures of the agents is based on our previous work [35], where a prototype of the scenario was developed, in order to have an initial decision architecture of the LW and kamikaze agents.

Many approaches to robotic decision-making have been developed; in modern applications such as autonomous cars, the agents are highly deliberative [36]. In these complex scenarios, decision-making is further subdivided into modules tailored to solve subtasks, which are referred to as behaviors [37]. Once this decomposition is performed, we need to orchestrate the execution of these behaviors. Many behavior selection mechanisms exist, with finite-state machines (FSMs) [38] and behavior trees (BTs) [39] being the most popular techniques. The design of these mechanisms is empirical and is mainly based on intuition, creativity, experience, and good practices [38]. Furthermore, this process is iterative, with the agent performance being evaluated by an expert or through statistics, in order to select the best agent architecture [39]. On the other hand, optimization methods and machine learning techniques may be employed to tune decision parameters [35] or to learn complete behaviors with no prior knowledge [40].

To start addressing Problem 2, we need the kamikaze agents to be functional via simplified but effective AI. Consequently, we selected the FSM technique to develop the decision-making module for the kamikaze. The FSMs are the most common model of computation for modeling decision-making architectures for simple AI [39], [41], due to its easy implementation and intuitive structure. In this technique, a state s represents a behavior for the agent. A FSM

can switch between behaviors in response to events. In Fig. 7, we present the decision-making developed for the kamikaze MAV, note that each MAV contains an identical decision module.

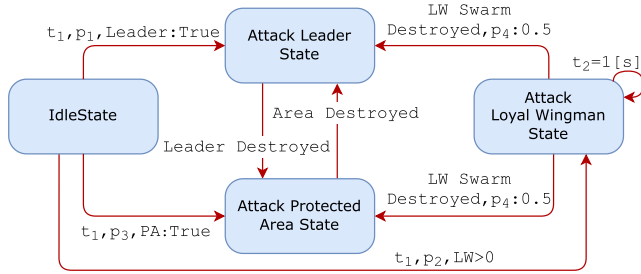


FIGURE 7. Decision-Making module for a kamikaze MAV using FSM.

In our solution, the kamikaze has a set \mathcal{K} of four behaviors, the initial behavior is *IdleState*, where the agent is idle for t_1 seconds and then selects a target to attack based on the probabilities p_1 , p_2 , and p_3 , which in this work are equally distributed probabilities used for each target type. Once a target is selected, the FSM will transition to the representative state of the selected target and will remain in that state until the agent is destroyed. The exception is *Attack Loyal Wingman*; this behavior targets the closest LW, and will select the closest LW every $t_2 = 1$ s, to avoid frequent target switching. Once there are no more LW, the FSM transitions to attack the leader or the protected area with a probability of 50% each.

In contrast, the behaviors to be executed by the LW MAV are more complex given their desired capabilities, consequently, its AI architecture requires further elaboration. To effectively solve Problem 2, we implement its decision-making module using the BT technique, due to its inherent advantages in comparison to FSMs, such as behaviors that are highly modular, reactive, and flexible to changes [14], [39]. We briefly present the basic BT framework used, but keep in mind that alternatives commonly employ extensions [14], [42]. A BT is composed by nodes of two types: *composite* or *leaf*. *Composite* nodes control the BT logic, while *leaf* nodes execute the behavior modules or condition checks. When executed, each node returns one of the following execution statuses: *Success*, *Failure*, or *Running*. Table 1 presents the return status logic of each node type.

Therefore, based on Problem 2 description, we can identify the modular behaviors for the decision-making of a LW, in Table 2. We present three movement behaviors: *Chase Threat*, *Go To formation*, and *Approach Formation*; and three offensive behaviors: *Vaporize Threat*, *Freeze Threat* and *Sacrifice Attack*.

Fig. 8 presents the behavior tree for a LW MAV using the behaviors described in Table 2. Note that every LW has an identical and independent copy of this decision-making module.

TABLE 1. Node types of a BT.

Node type	Success	Failure	Running
<i>Selector</i>	If one child succeeds	If all children fail	If one child returns running
<i>Sequence</i>	If all children succeeds	If one child fails	If one child returns running
<i>Parallel</i>	If N children succeeds	If $M-N$ children fail	If all children return running
<i>Action</i>	Upon completion	When impossible to complete	During completion
<i>Condition</i>	If true	If false	Never
<i>Inverter (Decorator)</i>	If Failure	If Success	-

TABLE 2. Behavior description for an LW MAV.

Behavior	Description
<i>Chase Threat</i>	The loyal wingman leaves the formation to pursue a threat in order to neutralize it.
<i>Go To Formation</i>	The loyal wingman returns to formation.
<i>Approach Formation</i>	The loyal wingman approaches the leader from a safe distance from the formation.
<i>Vaporize Threat</i>	Performs an attack using the vaporizer gun.
<i>Freeze Threat</i>	Slows down a kamikaze for a limited period of time.
<i>Sacrifice Attack</i>	When there is no ammunition left, the LW will disable the threat by sacrificing itself as a weapon of last resort.

This decision-making module will run by default the *Go To Formation* behavior, where the LW keeps a flying formation surrounding the leading drone while there are no threats identified. Whenever a kamikaze drone crosses the engagement range or the LW is allocated to a task, it enters into the *Chase Threat* behavior, for which the vaporizer weapon must be available. In this case, the agent leaves formation to pursue the threat and tries to neutralize it using available weapons. Once a kamikaze is in range of use of the weapons, the LW selects a neutralization method. The selected method may be the vaporizer gun, the freezing gun or a sacrifice as last resort when there is no remaining ammo. A weapon is *available* if there is still ammunition and if it is not cooling down after being fired. Note that the engagement distance and the range of use of both weapons are decision-making parameters, with the latter being less or equal than the nominal range of the weapons, thus leaving the use of weapons as a strategy criterion.

The neutralization strategy embedded in this BT is to first select the closest threat within range of his freezing gun (mid-range) and then try to freeze it. Once the threat is slower, it is safer to approach it, then the LW uses its vaporizer gun (short-range) to eliminate the threat. Note that the respective gun must be available at the time that this weapon needs to be used. The *SacrificeAttack* behavior is executed when there are no remaining ammo in both weapons, it causes the LW to collide with an imminent threat, thus using its body-frame as a protective weapon. Immediately after the threat is eliminated or becomes out of range, the LW returns to formation, thus ensuring the protection of the leader. In order to join the formation, the LW must following the sequence present in Fig. 9. Initially, the LW must first approach the formation from a safe distance $d_f \in \mathbb{R}_{>0}$ and send a request to the leader asking for permission to rejoin. The leader receives

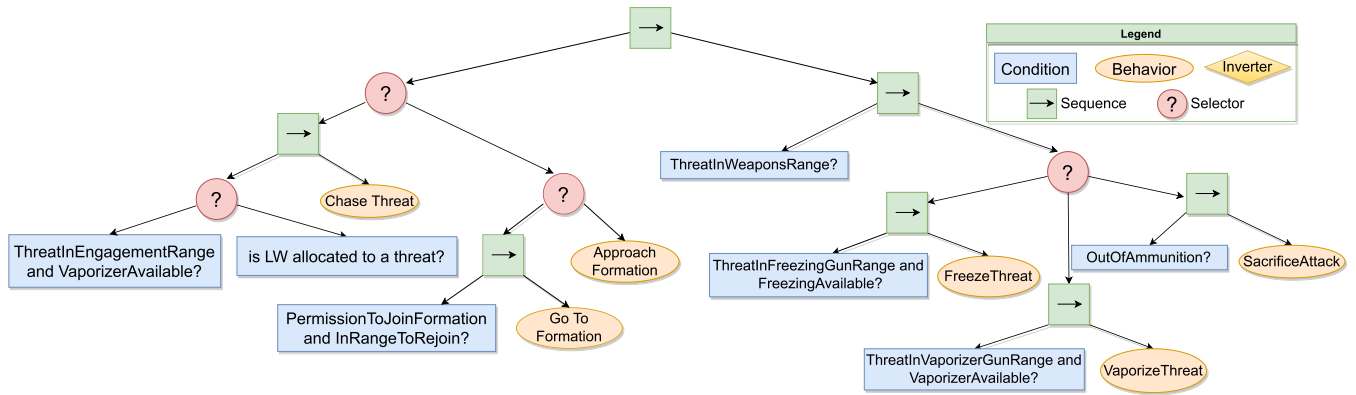


FIGURE 8. Behavior tree for the loyal-wingman MAV.

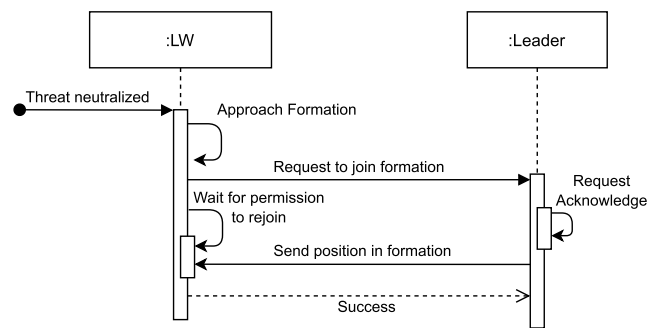


FIGURE 9. Join formation sequence diagram.

and processes the request, allowing or not the entry into formation. The LW waits for the permission to be granted and for the coordinates in the formation. The safe distance d_f is a parameter.

Note that the decision-making modules described in this section make decisions based only on its internal state and the states of the other agents present in the scenario, which are received through messages from the ground assets. Also, it is important to reiterate that the leader is capable of overriding the decisions made by the module of a specific LW, by either enforcing the loyal wingmen to return to formation, or selecting a specific LW to engage a selected kamikaze.

C. FORMATION DEFINITION

In this work, we consider that the leader defines the formation pattern as an uniform-circular formation along the local horizontal plane with a desired radius $r_f > 0 \in \mathbb{R}_{>0}$, as shown in Fig. 10. Given that there are a total of $n_f \in \mathbb{Z}_{>0}$ LW in formation, the positions occupied by the loyal wingmen are defined by the set

$$P_f \triangleq \left\{ r_g^{\ell/s} + r_f D_3(j\alpha + \phi) e_1 \right\}_{j=1}^{n_f}, \quad (22)$$

where $D_3(\cdot) \in SO(3)$ is the rotation matrix around \vec{z}_g , $\alpha = 2\pi/n_f$, and $\phi \in [-\pi, \pi)$ is an offset angle that can be used to rotate the formation around the leader. The

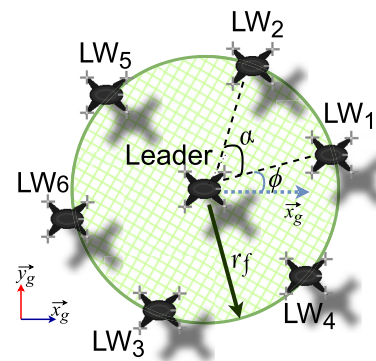


FIGURE 10. Uniform-circular formation pattern.

assignment of the LW to the available positions in formation is sequential, based on the order of the acknowledged request to join the formation by the leader, and not necessarily match the vehicle ID.

D. LW-KAMIKAZE TASK ALLOCATION

This subsection addresses the solution of Subproblem 2.3, which employs the Hungarian algorithm for the task assignment between the LW.

As seen in Fig. 11, the workers in our context are interpreted as the LWs available and the tasks are the neutralization of the kamikaze threats. The distance of the loyal wingman to a threat can be described as the euclidean distance $d(\mathbf{p}_i, \mathbf{q}_j) \triangleq \|\mathbf{p}_i - \mathbf{q}_j\|$, where $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_j \in \mathbb{R}^3$ are the current position of the i th LW and the j th kamikaze threat, respectively. So the cost of the i th LW to neutralize the j th kamikaze is

$$c_{i,j} \triangleq \begin{cases} d(\mathbf{p}_i, \mathbf{q}_j) + A_v P, & d(\mathbf{p}_i, \mathbf{q}_j) \leq d_a, \\ +\infty, & d(\mathbf{p}_i, \mathbf{q}_j) > d_a, \end{cases}$$

where $A_v \in \{1, 0\}$ is a variable that is 1 if the LW weapons are available, and 0 otherwise. The constant P represents an additional penalty cost to assign a LW that does not have a available weapon, as it is important to avoid the allocation of

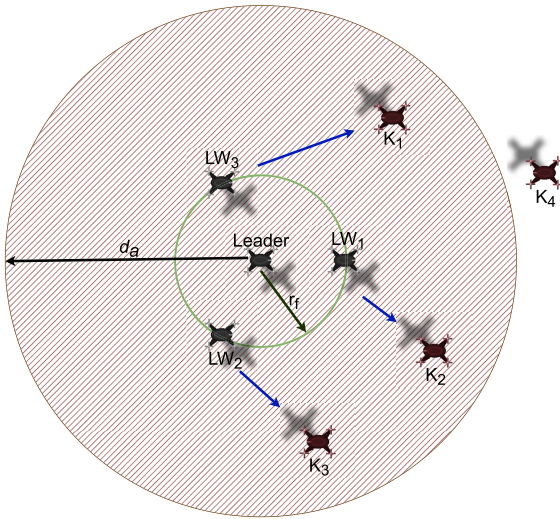


FIGURE 11. Diagram of LW-Kamikaze Task allocation problem.

unavailable LWs. Lastly, the engagement range is represented by the radius $d_a > 0$, and the threats outside this range will not be considered during the assignment.

The execution of the Hungarian algorithm outputs the lowest cost path through the C matrix, representing the optimal LW-Kamikaze assignment. In the example provided in Fig. 11 we have the optimal LW-Kamikaze task allocation indicated by the blue arrows, note that the threat k_4 is not considered, since it is outside d_a .

E. COOPERATION AND SELFISHNESS BEHAVIORS

The CEC is an emerging systems-of-systems capability in which multiple systems coordinate their actions through a network to improve their abilities to effectively perform a given task [1]. Highly developed systems of cooperation and mutual support can be found in different sorts of systems, a general consensus is that cooperation is in general advantageous for the group of cooperators as a whole, even though it may curb some individual’s resources [43].

Altruism means that a system performs actions that increase the fitness of another system. The term *Weak* altruism can be defined as behavior that benefits more to another individual than to the individual performing the behavior. A *Strong* altruism denotes behavior that benefits others, but at one’s own cost [44]. Both are common and necessary in those highly cooperative systems, which often have a collective organization with full division of labor, including individuals who are only defended by others, i.e., the MAV leader, or who are prepared to sacrifice themselves for the defense of others, i.e., the LW who sacrifices itself in extreme cases to ensure the protection of the leader. These kind of systems are found in certain species of insects, and in human society [44].

On the other hand, the term *Selfishness* means that the system will only perform actions that increase its own fitness [43], naturally selected systems will not only be

selfish, since they try to optimize their own fitness, but they will also tend to avoid altruism. Thus, one’s own fitness is indirectly reduced by altruistic behavior.

In our context, we can illustrate the cooperative behavior as seen in the following Fig. 12. In the frame (a) we have a LW_1 allocated by the leader to neutralize the threat K_4 , so here we have the LW_1 acting in an altruistic way for the benefit of the defense of the leader, also note the existence of another threat called K_1 in this frame. During the displacement towards the threat K_4 , seen in frame (b), the threat K_1 enters the selfish engagement radius (d_s), in this case the LW_1 identifies the existence of a threat within d_s and starts behaving selfishly. In this case the LW_1 decides to temporarily overwrite the command given by the leader, and neutralize K_1 which presents a great risk to itself. Finally, we have the frame (c), demonstrating that once the threat K_1 has been destroyed, the LW_1 starts to follow its altruistic behavior again, where it will neutralize the threat commanded by the leader.

Therefore, as demonstrated, the developed decision-making can behave both altruistically and selfishly. The altruistic behavior being the leader’s defense, even if it leads to his own destruction. And the selfish behavior, where he overwrites the leader’s assignment, thus ceasing to cooperate, and starts to chase the neutralization of the highest risk threat, which is the closest to him. Once it neutralizes the highest risk threat, it goes back to cooperate with the system as a whole, seeking to neutralize the threat allocated by the leader.

V. DECISION-CONTROL INTERFACE

As we have two main layers of software, it is necessary to define a communication interface between such modules, i. e., how the decision commands are translated into control commands for the MAVs.

As defined in Fig. 7, the kamikaze MAVs have four behaviors: *idle State*, *Attack Leader State*, *Attack Loyal Wingman State*, and *Attack Protected Area State*. The HL-LL interface of the kamikazes translates these behaviors only into target position commands since the kamikaze behaviors does not demand attitude changes.

The *idle State* demands the kamikaze to stay stationary in its current position, and to do so the target position informed to the guidance algorithm must be equal to the kamikaze position, i.e.,

$$\check{r}_g^{j/g}(t) = r_g^{j/g}(t),$$

where $j \in \mathcal{I}_k$.

On the other hand, *Attack Leader State*, *Attack Loyal Wingman State*, and *Attack Protected Area State* can be executed by generating the target position as

$$\check{r}_g^{j/g}(t) = r_g^{k/g}(t), \tag{23}$$

where k is the ID of the desired target.

As shown in Table 2, the LW has four movement behaviors: *Chase Threat*, *Go To formation*, *Approach Formation*, and *Sacrifice Attack*.

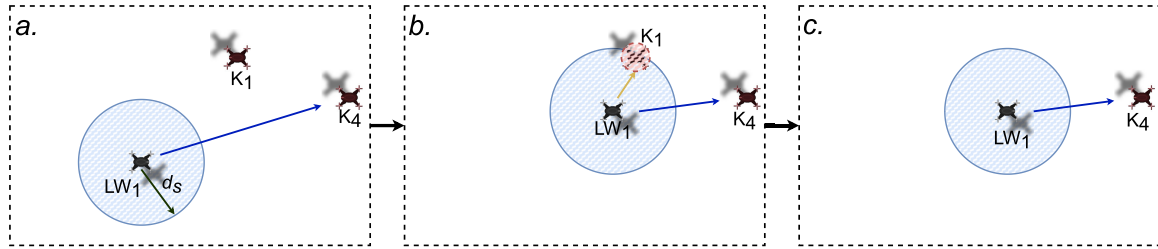


FIGURE 12. Altruistic and Selfish behavior sequence.

To execute the *Go To Formation*, the HL-LL interface of the LW informs the target position to the guidance algorithm as

$$\check{r}_g^{i/g}(t) = r_g^{\ell/g}(t) + \check{r}_g^{i/\ell},$$

where $i \in \mathcal{I}_m \setminus \ell$ and $\check{r}_g^{i/\ell} \in \mathbb{R}^3$ is a target position relative to the leader given by the HL decision-making layer.

Similarly, to execute *Approach Formation*, the target position is calculated as

$$\check{r}_g^{i/g}(t) = r_g^{\ell/g}(t) + \check{r}_g^{i/\ell} + e_3^T d_f.$$

Since the formation is defined on the local horizontal plane, the LW approaches the formation by taking a vertical offset relative to its target position in formation for simplicity.

To execute the *Chase Threat*, the target position informed to the guidance algorithm of the LW must be set as an offset relative to the target kamikaze position, *i.e.*,

$$\check{r}_g^{i/g}(t) = r_g^{k/g}(t) - d_t \frac{r_g^{k/\ell}(t)}{\|r_g^{k/\ell}(t)\|}, \quad (24)$$

where $i \in \mathcal{I}_m \setminus \ell$, $k \in \mathcal{I}_k$ is the ID of the target kamikaze, and $d_t \in \mathbb{R}_{>0}$ is the distance to be kept from the target. The *Sacrifice Attack* is executed by setting $d_t = 0$.

VI. COMPUTATIONAL ARCHITECTURE

This section describes the computational architecture used to implement the proposed scenario. A simulation framework is built as a proof of concept for the development of the proposed techniques. The infrastructure is divided into two PCs, as seen in Fig. 13. The first is responsible to compute the LL control, physics, 3D visualization, and interface with the human pilot through a joystick. The second one is responsible to run the HL decision-making modules of the agents being simulated. The computers communicate on a local network using the Robot Operating System 2 (ROS 2) middleware which is built on an open-source data distribution standard (DDS) middleware that provides features such as discovery, serialization, and transportation.¹

The LL control, physics, and interface with the human pilot through a joystick are coded in MATLAB. The 3D simulator was developed using the Unity engine to allow a proper visualization of the proposed scenario. On the other hand, the

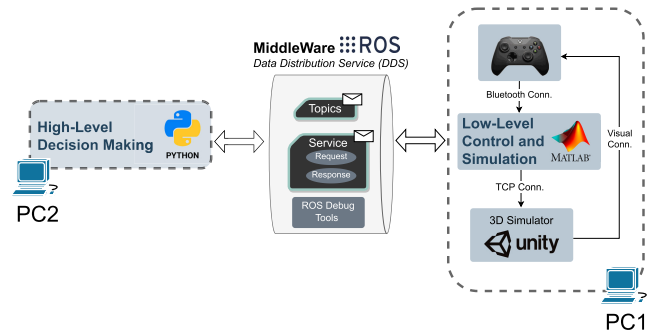


FIGURE 13. Diagram of the computational implementation.

decision-making algorithms were implemented in the Python programming language, due to its large community, support, and availability of AI libraries, especially for machine learning.

The data communication interface between the HL decision-making layer and the LL control layer is implemented using the framework provided by ROS 2 [45]. It is based on topics used to publish and receive customized messages in a publish-subscribe pattern that allows the layers to share important information asynchronously. The customized messages contain information regarding the action commands from the HL decision-making module and the status update of the vehicles calculated by the LL control and simulation sub-module, as described in Table 3. For ease of implementation, each vehicle is identified by a unique ID. The LL control and simulation sub-module also sends unidirectional information containing the pose of the vehicles via socket to the 3D simulator that works as a visual feedback to the pilot.

VII. SCENARIO SIMULATION AND RESULTS

In this section, we describe a series of simulations used to evaluate the overall method. Subsection VII-A shows the results of a formation flight simulation. Subsection VII-B presents the qualitative results of the HL decision making. Subsection VII-C presents a Monte Carlo simulation. Lastly, Subsection VII-D shows the computational burden of the overall simulation.

In all simulations, the MAVs are non-planar fully actuated octocopters, as the one depicted in Fig. 1, with a total mass

¹ROS2 Documentation: <https://www.ros.org>

TABLE 3. Table of commands.

Type of command	Command	Parameters
Movement	GO_TO_POSE	Target position and attitude.
	SET_FORMATION_POSE	Leader ID and relative position and attitude w.r.t. to the leader.
	CHASE	Threat ID to be pursued and neutralized.
Weapon	SHOOT	Orders the firing of a specified gun on a desired target

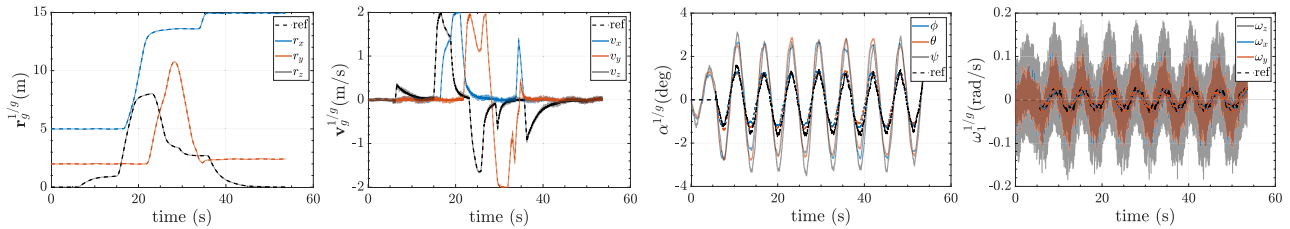


FIGURE 14. Position and attitude tracking performance of the LW 1. The symbol $\alpha^{1/g}$ is the Euler angles in the 1-2-3 sequence corresponding to $D^{1/g}$.

of 1 kg, arm length of 0.5 m, and inertia matrix equal to $diag(0.015, 0.015, 0.015)$ kgm^2 . Moreover, the MAVs are subject to a sinusoidal force disturbance with the same amplitude of 0.6 N but with different phase shifts, and to a sinusoidal torque disturbance with the same amplitude of 0.05 Nm but also with different phase shifts.

The LL control and simulation, coded in MATLAB, use the Euler integration method and runs with a frequency of 100 Hz. The actuator dynamics are modeled by first-order differential equations with a time constant of 0.01 s. The force and torque coefficients of the rotors are 1.2838×10^{-5} N/(rad/s)² and 3.0811×10^{-7} Nm/(rad/s)², respectively. It is worth mentioning that the adopted model is typically used in the MAV literature to describe the set consisting of a brushless motor, an electronic speed controller, and a propeller [46].

A. LOW-LEVEL CONTROL SIMULATION

To evaluate the LL control methodology, we simulate four LW flying in a circular-uniform formation around the remote-controlled leader in a scenario without kamikazes. Since the MAVs use the same flight control law, we evaluate the tracking performance of the LL control method by only plotting the tracking errors for the LW 1. The simulation of this flight can be seen in this video.²

Fig. 14 shows the position and attitude tracking performance of the LW 1. It can be seen that the vehicle performs an accurate position and velocity tracking, being the chattering [47], which is a high-frequency oscillation around the sliding manifold caused by unmodeled phenomena, such as time discretization, actuator dynamics, and sensor noise, more evident in the velocity components. It can also be seen that the attitude variables have a chattering with a magnitude smaller than two degrees in each component. Moreover, note that at the time instant 5.6 s, the attitude

chattering is amplified since at this instant the vehicle is called to formation, i.e., it starts receiving the leader attitude and angular velocity references that also present chattering.

B. HIGH-LEVEL DECISION MAKING QUALITATIVE RESULTS

In this section, we describe the qualitative results of a series of simulations conducted using the proposed computational architecture for 4 LW against 2 kamikazes. This simulations were used to test the effectiveness of the overall method, i.e., the effectiveness of the proposed LL control method and the LW BT architecture. The results described in this subsection can also be seen on this video.³ Moreover, the simulations were used to empirically find an effective LW decision-making architecture, while developing the behaviors and tuning the decision-making parameters. Subsequently, we demonstrate operating sequences found during the execution of the scenario.

First, in Fig. 15, we see the simulation result of the case where a kamikaze threat is identified and neutralized. Initially, in frame (a) we see the MUM-T in formation and an identified kamikaze threat in red. The LW MAV closest to the threat leaves the formation and chases the kamikaze by executing the *chase threat* behavior. As a result, all remaining LW MAV members in formation will readjust to new positions defined by the leader, so no gaps for attacks are present.

Consequently, in frame (b), we have the LW that left the formation attacking the kamikaze with the freezing gun, this strategy aims to facilitate the approach for neutralization since this weapon considerably reduces the kamikaze’s speed. Now in frame (c) the LW gets even closer to the frozen kamikaze and uses its vaporizer gun to effectively destroy it. Consequently, we see in frame (d) that the kamikaze is destroyed and the LW returns to formation, initially the LW

²<https://youtube.com/clip/UgkxHcMf1eOj>

³<https://youtu.be/YQjDRBW-Onk>

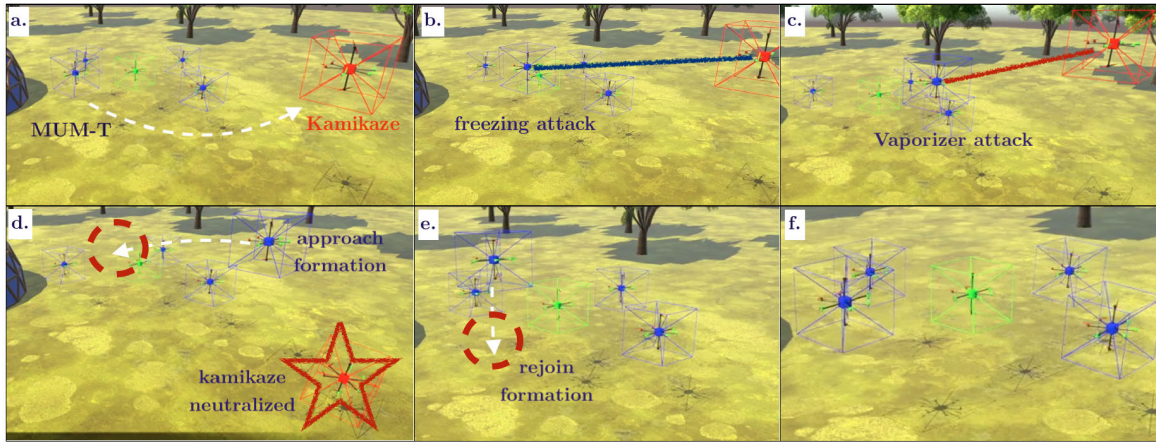


FIGURE 15. Demonstration sequence of engagement to a kamikaze threat, in red. The leader is seen in the center, in green, and four loyal wingman in blue.

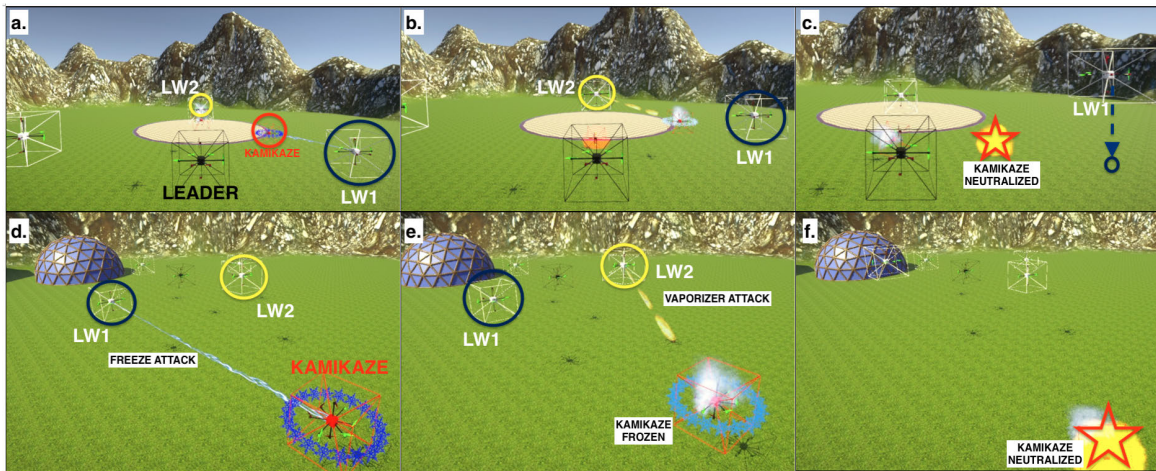


FIGURE 16. Demonstration of a cooperative neutralization sequence. In this simulation, the leader is in seen in black. The LW1 are represented in the blue circle, LW2 in the yellow circle and the kamikazes in red.

must approach the formation from a safe distance executing the *approach formation* and request re-entry for the leader, who must allow its entry and readjust the formation with the addition of this agent. In frame (e) we see the LW successfully re-entering the formation and sequentially executing the behavior *go to formation*. Finally, in frame (f) we see all the agents keeping formation after the success of the mission.

Furthermore, in Fig. 16, we observe two sequences of cooperative neutralization that emerged during the simulations, where two loyal wingmen performed tasks together to neutralize kamikazes. In the first sequence, frames (a)-(c), we have LW1 freezing the kamikaze from a longer distance, consequently, we observe LW2 in (b), which also identified the same threat, neutralizing it using the vaporizer gun. In the second sequence, frames (d)-(f), we see a similar case from a different view. In both cases, the participating LW1 and LW2 take complementary actions to eliminate a common threat.

Now we present the sequence leading to the sacrifice of an LW, in Fig. 17. The case begins in frame (a) with three loyal wingmen in formation, after LW4 is destroyed. In frame (b), we observe the LW1, which only has one cartridge remaining in its weapons, engaging two kamikazes, note that after the neutralization of the first kamikaze, it collides with the other one present in the attack, thus eliminating two threats. In frame (d) we have LW1 out of ammo and starting its sacrifice attack after a new threat enters the area. In frame (e) we have the successful elimination of the kamikaze, and consequently, the leader readjusts the relative position in formation of LW2 and LW3. Finally, in frame (f) we have a new MUM-T formation composed only by the remaining LW2 and LW3.

Unexpected behaviors can emerge, as illustrated in Fig. 18, where we observe two kamikazes heading toward the leader, in frame (a). In the sequence, in frame (b), we have LW1 identifying and freezing the closest threat K1. As a consequence, we observe that K1 collides with K2, since K1

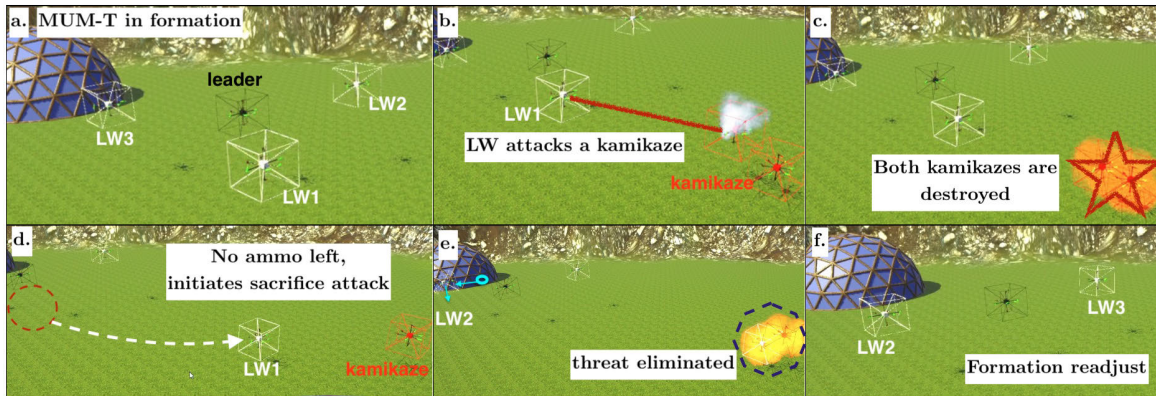


FIGURE 17. Demonstration of the sacrifice neutralization sequence. In this simulation, the leader is in seen in black. The LW are represented in white and the kamikazes in red.

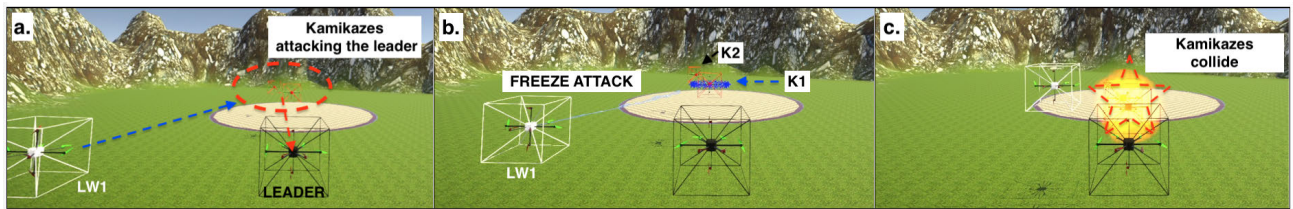


FIGURE 18. Freezing attack being used as a neutralization method. In this simulation, the leader is seen in black. The LW1 in white and the kamikazes K1 and K2 in red.

has been frozen and K2 which is closely following K1 cannot avoid the collision in time. In this way, we can highlight this emergent behavior as an indirect neutralization method.

C. MONTE CARLO SIMULATIONS

The effectiveness of the overall method is evaluated using a Monte Carlo simulation with 122 iterations. These simulations are performed considering 4 LW against 3 kamikazes (that respawn once neutralized) with the protected area located at the origin and the leader hovering at the point (5, 0, 5) m. The LW have a maximum speed of 1.5 m/s, being the kamikazes 50% faster. The freezing and vaporizer guns have an ammo of 10, a cooling down time of 1 s, and a hit probability of 95%. The freezing gun can reduce the kamikaze speed by half during 5 s.

Fig. 19 shows the Monte Carlo simulation results. It can be seen that, on average, the survival time is 169.3 s and the number of kamikazes destroyed is 34, without considering the sacrifice attack behavior, which is an excellent result since the MUM-T can only directly neutralize a maximum of 40 kamikazes given that each LW is equipped with a vaporizer gun with an ammo of 10. This corresponds to 85% of the total neutralizing capability.

D. COMPUTATIONAL TIME

To evaluate the computational burden of the overall simulation, we conducted a typical run using the same parameters adopted in Subsection VII.C. For this run, both the LL and the

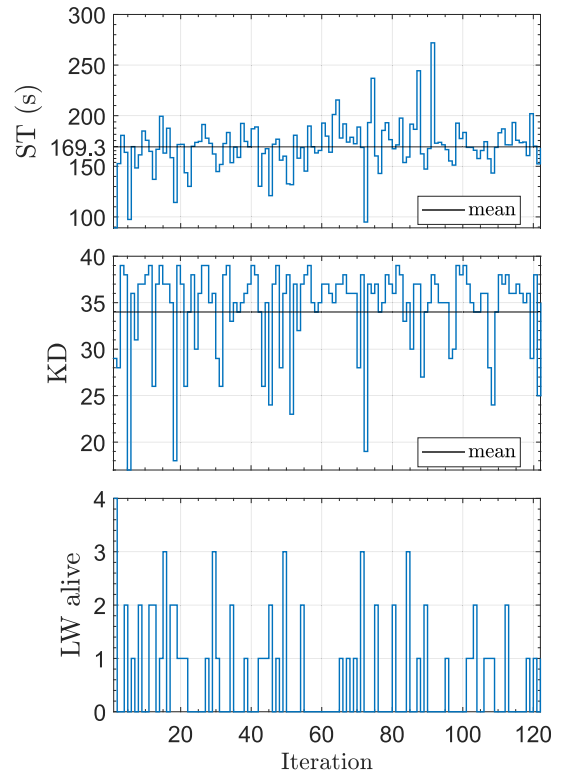


FIGURE 19. Monte Carlo simulation results. ST stands for survival time and KD stands for kamikazes destroyed.

HL modules for all the MAVs were implemented in the same computer, which is a laptop equipped with a 12th generation

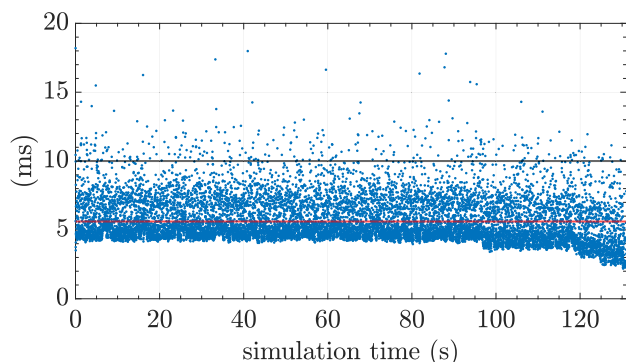


FIGURE 20. Overall simulation iteration time. Legend: ● iteration time, — iteration time average, and — adopted sampling time.

Intel Core i7-12700H, with 2.30 GHz and 16 GB of RAM. The total simulation time, from the beginning to the leader destruction, was about 131 s. In only 3.4% of a total of 13100 time steps, the sampling time of 0.01 s was exceeded.

Further, similar to the computational analysis done in [48], Fig. 20 shows a plot of the iteration time over the simulation time, together with the respective average and the adopted sampling time. From this plot, we can see that, in fact, in most of the iterations, the adopted sampling time is not violated. It is worth mentioning that, these results are satisfactory since real time is reasonably achieved even using a code that is not optimized for embedded implementation.

VIII. CONCLUSION

This paper evaluated the concept of CEC applied to a MUM-T. To this end, we proposed a defensive scenario where loyal wingman MAVs cooperate to defend a manned leader and protect a critical infrastructure from a swarm of kamikazes with explosive capabilities. To deal with such threats the loyal wingmen have the capability to engage and neutralize the threats, using two idealized weapons, a vaporizer gun representing a short-range weapon capable of neutralizing a threat with a single shot, and a freezing gun, a non-lethal weapon of mid-range, capable of slowing down the threat. To reduce the complexity of this problem, all the vehicles are modeled as fully actuated MAVs, and the problem is broken down into two parts, one involving an LL control layer, and another involving an HL layer of intelligent software capable of making autonomous and decentralized decisions. The LL control designs a joint attitude-position control law using a first-order sliding mode control that guarantees the closed-loop system robustness with respect to model uncertainties and disturbances. On the other hand, the decision-making for loyal wingman agents was developed using the behavior tree technique and reduced to modular behaviors, such as: chasing a threat, remaining in protective formation, approaching formation to initiate rejoin procedure, and attack behaviors using the available weapons or using the sacrifice attack in the extreme case where there are no more weapons available. The method

is analyzed using a high-fidelity 3D simulator and shown to be effective. In future works, we plan to increase the complexity of the kamikazes' behaviors in order to improve the difficulty of the proposed scenario and investigate different formation patterns. We also plan the development of consensus algorithms using a world model that is not synchronized among the MUM-T agents. Moreover, more complex scenarios can be defined involving different types of threats and drones.

This paper reported a low technology readiness level (TRL) research project, which is expected to move to a higher TRL in future works. Moreover, in an immediate future phase, this project will focus on state estimation (navigation and tracking) as well as on distributed hardware-in-the-loop simulation.

ACKNOWLEDGMENT

Jorge A. Ricardo Jr. acknowledges the Coordination of Superior Level Staff Improvement (CAPES) and EMBRAER S. A., for the doctorate scholarship under the Academic-Industrial Graduate Program (DAI) and Luiz Giacomossi acknowledges EMBRAER S. A., for his scholarship.

REFERENCES

- [1] APL-Team, "The cooperative engagement capability," in *Johns Hopkins APL Tech. Dig.*, 1995, vol. 16, no. 4, p. 377.
- [2] D. Xing, Z. Zhen, and H. Gong, "Offense-defense confrontation decision making for dynamic UAV swarm versus UAV swarm," *Proc. Inst. Mech. Eng. G, J. Aerosp. Eng.*, vol. 233, no. 15, pp. 5689–5702, Dec. 2019, doi: 10.1177/0954410019853982.
- [3] Y. Gao and D. Li, "Unmanned aerial vehicle swarm distributed cooperation method based on situation awareness consensus and its information processing mechanism," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105034, doi: 10.1016/j.knsys.2019.105034.
- [4] C. Walton, I. Kaminer, Q. Gong, A. H. Clark, and T. Tsatsanis, "Defense against adversarial swarms with parameter uncertainty," *Sensors*, vol. 22, no. 13, p. 4773, Jun. 2022, doi: 10.3390/s22134773.
- [5] B. Qin, D. Zhang, S. Tang, and M. Wang, "Distributed grouping cooperative dynamic task assignment method of UAV swarm," *Appl. Sci.*, vol. 12, no. 6, p. 2865, Mar. 2022, doi: 10.3390/app12062865.
- [6] J. Yoo, H. Seong, D. H. Shim, J. H. Bae, and Y.-D. Kim, "Deep reinforcement learning-based intelligent agent for autonomous air combat," in *Proc. IEEE/AIAA 41st Digit. Avionics Syst. Conf. (DASC)*, Sep. 2022, pp. 1–9.
- [7] C. Humphreys, R. Cobb, D. Jacques, and J. Reeger, "Optimal mission path for the uninhabited loyal wingman," in *Proc. 16th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Jun. 2015, p. 2792.
- [8] F. Santos, M. A. Garratt, and S. G. Anavatti, "Visual-Inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 260–275, Jan. 2017.
- [9] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1279–1291, Sep. 2012.
- [10] L. Giacomossi, J. A. Ricardo Jr., J. F. Brancalion, M. R. Maximo, and D. A. Santos, "Cooperative threat engagement with heterogeneous drone swarms," in *Proc. Congr. Int. Council Aeronaut. Sci.*, 2022, pp. 1–15.
- [11] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3261–3266.
- [12] D. A. Santos and J. A. Bezerra, "On the control allocation of fully actuated multirotor aerial vehicles," *Aerosp. Sci. Technol.*, vol. 122, Mar. 2022, Art. no. 107424.
- [13] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3475–3482, doi: 10.1109/ICRA.2011.5980408.

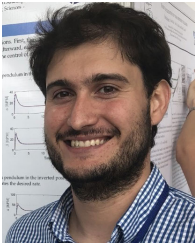
- [14] M. Iovino, E. Scukins, J. Styruð, P. Ögren, and C. Smith, "A survey of behavior trees in robotics and AI," *Robot. Auto. Syst.*, vol. 154, Aug. 2022, Art. no. 104096.
- [15] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics*, vol. 52, nos. 1–2, pp. 7–21, Feb. 2005.
- [16] A. L. Silva and D. A. Santos, "Fast nonsingular terminal sliding mode flight control for multirotor aerial vehicles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 6, pp. 4288–4299, Dec. 2020.
- [17] J. A. Bezerra and D. A. Santos, "On the guidance of fully-actuated multirotor aerial vehicles under control allocation constraints using the receding-horizon strategy," *ISA Trans.*, vol. 126, pp. 21–35, Jul. 2022.
- [18] J. A. Ricardo Jr. and D. A. Santos, "Smooth second-order sliding mode control for fully actuated multirotor aerial vehicles," *ISA Trans.*, vol. 129, pp. 169–178, Oct. 2022.
- [19] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control* (Space Technology Library). New York, NY, USA: Springer-Verlag, 2014.
- [20] D. A. Santos and C. M. Lagoa, "Wayset-based guidance of multirotor aerial vehicles using robust tube-based model predictive control," *ISA Trans.*, vol. 128, pp. 123–135, Sep. 2022, doi: [10.1016/j.isatra.2021.12.002](https://doi.org/10.1016/j.isatra.2021.12.002).
- [21] S. V. Dragunov and V. I. Utkin, "Sliding mode control in dynamic systems," *Int. J. Control*, vol. 55, no. 4, pp. 1029–1037, 1991.
- [22] Y.-S. Lu and J.-S. Chen, "Design of a global sliding-mode controller for a motor drive with bounded control," *Int. J. Control*, vol. 62, no. 5, pp. 1001–1019, Nov. 1995, doi: [10.1080/00207179508921579](https://doi.org/10.1080/00207179508921579).
- [23] V. Utkin and J. Shi, "Integral sliding mode in systems operating under uncertainty conditions," in *Proc. 35th IEEE Conf. Decis. Control*, vol. 4, Dec. 1996, pp. 4591–4596, doi: [10.1109/CDC.1996.577594](https://doi.org/10.1109/CDC.1996.577594).
- [24] J. A. Ricardo and D. A. Santos, "Robot guidance and control using global sliding modes and acceleration velocity obstacles," in *Proc. 16th Int. Workshop Variable Struct. Syst. (VSS)*, Sep. 2022, pp. 41–46.
- [25] R. E. Burkard and E. Çela, "Linear assignment problems and extensions," in *Handbook of Combinatorial Optimization*. Boston, MA, USA: Springer, 1999, pp. 75–149, doi: [10.1007/978-1-4757-3023-4_2](https://doi.org/10.1007/978-1-4757-3023-4_2).
- [26] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.
- [27] D. A. D. Santos and A. Cunha, "Flight control of a hexa-rotor airship: Uncertainty quantification for a range of temperature and pressure conditions," *ISA Trans.*, vol. 93, pp. 268–279, Oct. 2019, doi: [10.1016/j.isatra.2019.03.010](https://doi.org/10.1016/j.isatra.2019.03.010).
- [28] V. I. Utkin, *Sliding Modes in Control and Optimization*. Berlin, Germany: Springer-Verlag, 1992.
- [29] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM J. Control Optim.*, vol. 38, no. 3, pp. 751–766, Jan. 2000.
- [30] J. A. Ricardo Jr., D. A. Santos, and T. R. de Oliveira, "Attitude tracking control for a quadrotor aerial robot using adaptive sliding modes," in *Proc. XLI Ibero-Latin-Amer. Congr. Comput. Methods Eng.*, Nov. 2020, pp. 1–7.
- [31] J. A. Ricardo Jr. and D. A. Santos, "Position guidance and control for fully actuated multirotor aerial vehicles in dynamic environments," in *Proc. XLIII Ibero-Latin-Amer. Congr. Comput. Methods Eng.*, 2022, pp. 1–7.
- [32] Y. Fang, J. Hu, W. Liu, Q. Shao, J. Qi, and Y. Peng, "Smooth and time-optimal S-curve trajectory planning for automated robots and machines," *Mechanism Mach. Theory*, vol. 137, pp. 127–153, Jul. 2019.
- [33] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, Jul. 1998, doi: [10.1177/027836499801700706](https://doi.org/10.1177/027836499801700706).
- [34] J. Van Den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," *IEEE Trans. Robot.*, vol. 23, no. 4, p. 834, May 2008, doi: [10.1109/TRO.2007.905255](https://doi.org/10.1109/TRO.2007.905255).
- [35] L. Giacomossi, S. S. Dias, J. F. Brancalion, and M. R. O. A. Maximo, "Cooperative and decentralized decision-making for loyal wingman UAVs," in *Proc. IEEE Latin Amer. Robot. Symp. (LARS)*, Oct. 2021, pp. 78–83.
- [36] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [37] T. P. Baldão, M. R. O. A. Maximo, and C. de Azevedo Castro Cesar, "Decision-making for 5x5 very small size soccer teams," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR) Workshop Robot. Educ. (WRE)*, Nov. 2020, pp. 1–6.
- [38] M. Buckland, *Programming Game AI by Example*. Burlington, MA, USA: Jones & Bartlett, 2004.
- [39] M. Colledanchise and P. Ogren, *Behavior Trees in Robotics and AI: An Introduction*. Boca Raton, FL, USA: CRC Press, 2018.
- [40] L. C. Melo, D. C. Melo, and M. R. O. A. Maximo, "Learning humanoid robot running motions with symmetry incentive through proximal policy optimization," *J. Intell. Robot. Syst.*, vol. 102, no. 3, pp. 1–15, 2021.
- [41] L. Giacomossi, F. Souza, R. G. Cortes, H. M. M. Cortez, C. Ferreira, C. A. C. Marcondes, D. S. Loubach, E. F. Sbruzzi, F. A. N. Verri, J. C. Marques, L. A. Pereira, M. R. O. A. Maximo, and V. V. Curtis, "Autonomous and collective intelligence for UAV swarm in target search scenario," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR), Workshop Robot. Educ. (WRE)*, Oct. 2021, pp. 72–77.
- [42] G. L. Silva, M. R. O. A. Maximo, and L. A. Pereira, "A minimalist open source behavior tree framework in C++," in *Proc. Latin Amer. Robot. Symp. (LARS), Brazilian Symp. Robot. (SBR), Workshop Robot. Educ. (WRE)*, Oct. 2021, pp. 306–311.
- [43] F. Heylighen, "Evolution, selfishness and cooperation," *J. Ideas*, vol. 2, no. 4, pp. 70–76, 1992.
- [44] D. T. Campbell, "The two distinct routes beyond kin selection to ultrasociality: Implications for the humanities and social sciences," in *The Nature of Prosocial Development: Theories and Strategies*. New York, NY, USA: Academic, 1983, pp. 11–41.
- [45] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3, Kobe, Japan, 2009, p. 5.
- [46] J. A. Bezerra and D. A. Santos, "Optimal exact control allocation for under-actuated multirotor aerial vehicles," *IEEE Control Syst. Lett.*, vol. 6, pp. 1448–1453, 2022, doi: [10.1109/LCSYS.2021.3110490](https://doi.org/10.1109/LCSYS.2021.3110490).
- [47] H. Lee and V. I. Utkin, "Chattering suppression methods in sliding mode control systems," *Annu. Rev. Control*, vol. 31, no. 2, pp. 179–188, Jan. 2007.
- [48] P. Dini and S. Saponara, "Processor-in-the-loop validation of a gradient descent-based model predictive control for assisted driving and obstacles avoidance applications," *IEEE Access*, vol. 10, pp. 67958–67975, 2022, doi: [10.1109/ACCESS.2022.3186020](https://doi.org/10.1109/ACCESS.2022.3186020).



JORGE A. RICARDO JR. was born in Rio de Janeiro, Brazil, in 1991. He received the B.S. degree in mechanical engineering from the Federal University of Rio de Janeiro, Rio de Janeiro, in 2017, and the M.S. degree in mechanical engineering from the Aeronautics Institute of Technology, São José dos Campos, São Paulo, Brazil, in 2020, where he is currently pursuing the Ph.D. degree in mechanical engineering.



LUIZ GIACOMOSSI JR. received the B.Sc. degree in control and automation engineering from the Federal University of Technology—Paraná, Brazil, in 2016, and the Ph.D. degree in software engineering from the Pontifical Catholic University of Argentina, Buenos Aires, CABA, Argentina, in 2020. He is currently pursuing the M.S. degree in electronic and computer engineering with the Aeronautics Institute of Technology, Brazil. He is also a member of the Autonomous Computational Systems Laboratory (LAB-SCA) and a Researcher with the Artificial Intelligence and Intelligent Systems Group, Mälardalens Universitet, Västerås, Sweden. His research interests include robotics, machine learning, decision-making, and simulation of UAVs.



JOÃO F. S. TRENTIN received the B.Sc. and D.Sc. degrees in mechanical engineering from the School of Engineering of Ilha Solteira, São Paulo State University (UNESP), in 2016 and 2020, respectively. He was a Postdoctoral Researcher with the Aeronautics Institute of Technology (ITA), from 2021 to 2022. He is currently a Product Development Engineer at EMBRAER S. A. His research interests include nonlinear dynamics, control, multirotor aerial vehicles, and sliding mode control.



MARCOS R. O. A. MAXIMO received the B.Sc. degree (Hons.) (summa cum laude) in computer engineering and the M.Sc. and Ph.D. degrees in electronic and computer engineering from the Aeronautics Institute of Technology (ITA), Brazil, in 2012, 2015, and 2017, respectively. He is currently a Professor at ITA, where he is also a member of the Autonomous Computational Systems Laboratory (LAB-SCA) and leads the robotics competition team—ITAndroids. His research interests include humanoid robotics, mobile robotics, dynamical systems control, and artificial intelligence.



JOSÉ F. B. BRANCALION was born in Piracicaba, São Paulo, Brazil, in 1973. He received the B.S. and M.S. degrees in electrical engineering from the University of São Paulo, São Paulo, in 1997 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the Aeronautics Institute of Technology, São José dos Campos, Brazil, in 2015. Since 2001, he has been a Product Development Engineering with the Research and Development Department,

EMBRAER S.A. He has published journal articles and conference papers in his areas of expertise. His research interests include high level information fusion processes and applications, multi-target tracking, multi-sensor data fusion, and key technologies for mobile and satellite communications. He has also served as a technical reviewer for some important international journals and conferences.



DAVI A. SANTOS received the Ph.D. degree in satellite attitude determination and fault-tolerant state estimation from the Instituto Tecnológico de Aeronáutica (ITA), Brazil, in 2011. He is currently a Control System Engineer with research experience in the aerospace field, with focus on aerial robotics and satellites. In 2014, he founded and has been leading a research laboratory on aerial robotics at ITA, where he holds a position of an Associate Professor, the Head of the Department of Mechatronics, and the Vice Chair of the Mechanical Engineering Division. He was an Invited Researcher at ESIEE Paris/Université Paris-Est, in 2013, and Penn State University, in 2018. His research interests include research and development related to guidance, navigation, and control of aerospace devices, such as satellites, rockets, launchers, and unmanned aerial vehicles.

• • •