Cooperative and Decentralized Decision-Making for Loyal Wingman UAVs

Luiz Giacomossi Jr.[1], Stiven Schwanz Dias[2], José Fernando Brancalion[3] and Marcos R. O. A. Maximo[1].

2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE) | 978-1-6654-0761-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/LARS/SBR/WRE54079.2021.9605468

***Abstract*—With the application of decision-making techniques, it is possible to automate many of the intelligent decisions of UAVs, leaving human commanders to focus on higher-level decisions in modern warfare scenarios, providing a crucial tactical advantage. This paper applies key methods to enable decentralized decision-making for autonomous combat UAVs, also known as loyal wingman (LW) drones. A simulator was developed to simulate to implement and analyze the behaviors of the so-called LW. We used as the main decision-making technique the construction of a behavior tree (BT) capable of providing decisions in a decentralized manner, and for parameters optimization, the PSO optimization algorithm was used. Our approach shows a promising result of threat elimination efficiency of approximately 93%.**

***Index Terms*—Cooperative engagement capability; manned-unmanned teaming; loyal wingman; Decision-Making techniques.**

I. Introduction

The term decision-making is regarded as the cognitive process resulting in the selection of a belief or a course of action among several possible alternatives [1]. Historically, the decision-making for the coordination of combat units was carried out by humans, involving hundreds to thousands of people. With the advances of communication and computing, modern decision-making enabled the so-called Cooperative Engagement Capability (CEC) among multiple networked agents [2], which can be applied to multiple manned and unmanned aircraft systems a.k.a Manned-unmanned teaming (MUM-T), to cooperatively engage and disable aerial threats.

The future vision combines a manned fighter commanding multiple unmanned aircraft, extending combat resources [3]. However, developing high-level decision-making techniques to CEC in real scale require a prohibitive amount of resources. The MUM-T includes the concept of the *Loyal Wingman* (LW) – an unmanned aerial vehicle (UAV) under the tactical command of a manned leader. Multiple autonomous LWs may be key to developing a new system that can act as a major force multiplier for existing human combat aircraft. Recently, drones have become more accessible and we have witnessed the emergence of UAVs in the most diverse applications [4], which are now suitable candidates for investigating MUM-T applications.

In this paper, we focus on the application of the loyal wingman concept with autonomous UAVs, allied with the MUM-T concept and with a combination of artificial intelligence (AI) techniques to increase their autonomy. It is possible to automate most of the actions and decisions done by combat UAVs using modern decision-making techniques, allowing human pilots to focus on the big tactical picture. We focus the research on its application in a simulated defensive scenario, with the objective of protecting a leading manned aircraft and a critical area from kamikaze drone attacks.

This research contributes by combining modern decision-making approaches with a swarm of LW UAVs to coordinate efficient defense actions in a cooperative, autonomous, and decentralized manner. We also contribute by introducing a novel defense scenario based on the loyal wingman concept, which we consider an interesting testbed for cooperative decision-making. The methodologies were combined with the creation of a UAV simulator, to make the LW behaviors and defense strategies easier to apply and evaluate.

The remaining of this document is organized as follows. In Section IIwe review key decision-making methods from literature. In the sequel, we describe the scenario of interest in Section III comprising a MUM-T and a kamikaze swarm. Later, in Section IV the simulator developed is presented. Then we proceed to the development of the agent behaviors in Section V. The experiments and their results are presented in Section VI, and we move to the discussion and results' analysis in Section VII. Lastly, Section VIII concludes and shares ideas for future work.

II. Decision-Making Architectures

In this section, the main methods used to control the behaviors for autonomous agents are presented. Many selection mechanisms exist, with finite-state machines (FSMs) [5] and behavior trees (BTs) [6] being the most popular.

Many approaches have been devised for robotic decision-making, in applications such as autonomous cars, the agents are highly deliberative [7]. In these complex scenarios, the decision-making is broken down into the so-called behaviors [8], which are modules tailored to solve subtasks. For example, Ogren [9] designed behaviors for a *Combat UAV* agent, the typical behaviors encompass: `EvasiveManeuvre`, `DoCombat`, `disEngage`, `FlyHome` and `Strike`.

Behavior selection mechanisms using FSMs or BTs are often designed by humans. The design is empirical and is

[1]Luiz Giacomossi Jr. and Marcos R. O. A. Maximo are with the Autonomous Computational Systems Lab (LAB-SCA), Computer Science Division, Aeronautics Institute of Technology, São José dos Campos, SP, Brazil `luiz.giacomossi@ga.ita.br`, `mmaximo@ita.br`.

[2]Stiven Schwanz Dias is with the Data Mining group, Mathematics and Computer Science department, Technische Universiteit Eindhoven, Eindhoven, North Brabant, Netherlands. `s.schwanz.dias@tue.nl`.

[3]José Fernando Brancalion is with the Technological Development Department, EMBRAER S.A., São José dos Campos, SP, Brazil `jose.brancalion@embraer.com.br`.

978-1-6654-0761-8/21/$31.00 ©2021 IEEE

TABLE I
NODE TYPES OF A BT.

Node type	Success	Failure	Running
Selector	If one child succeeds	If all children fail	If one child returns running
Sequence	If all children succeeds	If one child fails	If one child returns running
Parallel	If N children succeeds	If M-N children fail	If all children return running
Action	Upon completion	When impossible to complete	During completion
Condition	If true	If false	Never
Inverter	If `Failure`	If `Success`	-

mainly based on experience, creativity, and good practices [6], [8]. Furthermore, the process is iterative, with the agent performance being evaluated by a human or through statistics in order to select the best agent configuration [8]. On the other hand, optimization methods and machine learning techniques may be employed to optimize parameters, such as thresholds used to define conditions for behavior switching [10], [11].

A. Finite State Machines

FSMs are the most common mathematical model of computation where the system can be in only one of a finite number of states at any given time [5]. *i.e* an FSM guarantees the permanence in a certain state, unless a transition is triggered. The developer is responsible for defining the behaviors (states) and the conditions that trigger transitions between behaviors.

The wide use of FSMs is due to their intuitive structure and ease of implementation. However, FSMs have scalability disadvantages with the addition of behaviors and transitions, so code maintenance is laborious [6]. Reusability is also an issue, making it unpractical for reusing behaviors in other projects.

B. Behavior Trees

The behavior tree (BT) approach is to encode behaviors that are modular and reactive [6]. Since most of the problems found in the FSM are easily handled by BTs, the method has surpassed the FSMs as the industry standard in AI games [9].

A BT framework [6] is composed by nodes, which can be composite or leaf. Composite nodes control the BT logic, while leaf nodes execute the behaviors or check conditions. When executed, each node returns a execution status: `Success`, `Failure`, or `Running`.

The types of composite nodes are: sequence, selector, parallel, and decorator. Table I shows the return status logic of each node type. Sequence nodes sequentially executes all their children in order, as long as they are successful. A selector is used when any child can perform the task, it selects the first child that is successful. A parallel node executes its children in parallel (at the same run time). A decorator (`Inverter`) node changes the execution status of its child, many types of decorator may exist, depending on the framework used. Conditional check nodes are used to check if a condition is satisfied.

The leaf nodes are implemented by the agent developer: they are the behaviors themselves or conditional checks, *e.g.* a behavior such as `DefendLeader` will keep returning `Running` while the behavior is executing. Then, if the threat attacking was neutralized, the node return `Success`. Otherwise, if the leader is destroyed, the node returns `Failure`.

III. SCENARIO OF INTEREST

In this section, we introduce a novel defensive scenario based on the loyal wingman concept. An approach to achieve a higher understanding of the context and enabling its assessment is through gamification [12].

A. Defensive scenario

There are two teams, a MUM-T composed of a remotely-piloted leading UAV, escorted by a number of LW UAVs, and a kamikaze swarm. The MUM-T must engage and disable kamikaze drones, which try to attack the protected area and the lead drone. The scenario is depicted in Fig. 1.

There are two teams, a MUM-T composed of a remotely-piloted leading UAV, escorted by a number of LW UAVs, and a kamikaze swarm. The MUM-T must engage and disable kamikaze drones, which try to attack the protected area and the lead drone.

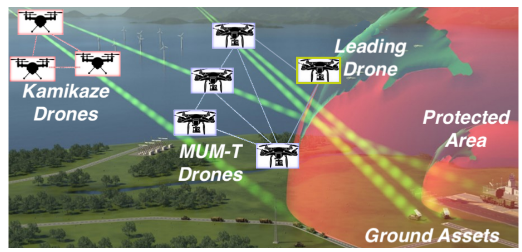

Fig. 1. Defense Scenario – MUM-T being attacked by kamikaze drones.

Therefore, the main objective and definition of the completion of the scenario are:

1) Mission: The LW shall protect both a protected area and the leading drone against the incursions of a kamikaze swarm.

2) Game over:

- The leading drone is hit by a single kamikaze drone, or;
- The critical area is hit by five drones, or;
- All loyal wingman were destroyed.

B. Description of the entities

The following list presents a description of the entities. Note that the parameters are represented using variable names in *snake_case* naming convention.

1) Manned-unmanned teaming (MUM-T):

- A single-player manually controls the leading drone;
- Loyal wingman agents are fully autonomous, and there are a fixed number of LW (`num_loyal_wingman`);
- The leading drone is in charge of the formation and for passing the formation coordinates to each LW.

2) MUM-T Weaponry:

- MUM-T drones have a mid-range freezing gun and a short-range vaporizer gun, both with a limited number of cartridges, given by `cartridges_vaporizing` and `cartridges_freezing`, respectively, and cooldown time of `freezing_cooldown` seconds and `vaporizer_cooldown` seconds, respectively;

- The freezing gun has a range of `freezing_range` meters and slows down the threat by 2/3 of its maximum speed for `time_frozen` seconds with a hitting probability of `freezing_hit_prob`. This weapon intends to make the decision space more complex and in practice the freezing dynamic is infeasible, given current technology;
- The vaporizer gun has a range of `vaporizer_gun_range` meters and destroys the threat with a hitting probability of `vaporizer_hit_prob`.

3) Kamikaze Agents:

- The kamikaze drone aims to strike the leading drone, protected area, or LW (we assume agents can identify them) and explode within range of contact;
- Kamikaze drones continuously appear within the range of ground-based radars, so up to `max_num_of_kamikaze` drones are present in the simulated scenario;
- Each Kamikaze drone will randomly select a target when created, and will not change the target until it is destroyed.

4) Ground Assets:

- The ground-based radars share situational awareness with MUM-T drones, *i.e.* a vector with the state (pose and velocity) of all entities. In simulation, in code it is represented as messages carried out between entities.

IV. Simulator

A simulator was entirely developed in *Python*, using for visual representation and user interaction the *Pygame* library. This simulator is intended to facilitate simulations through an easy-to-use, easy-to-modify, open-source code. The implementation of the defensive scenario focuses on the evaluation of decision-making techniques, abstracting a simulation with realistic physics. In this section, we describe the techniques used for the simulator creation. The interested reader can find further implementation details in Github[2].

A. Position Controller

Drones must be able to move at desired coordinates, for this purpose, a simplified $P+V$ controller was used, which computes the force to be applied to the drone rotors, given by

$$\overrightarrow{f} = K_p(\mathbf{p}_r - \mathbf{p}) - K_v\mathbf{v}, \tag{1}$$

where the force ($\overrightarrow{f}$) is proportional (K_p) to the difference in position ($\mathbf{p}$) with the desired coordinates ($\mathbf{p}_r$) and is proportional (K_v) to the current velocity ($\mathbf{v}$) of the drone.

Note that the force must be limited to the maximum force that can be applied by the rotors. In our simulator, the acceleration of the model is the force divided by the mass of the UAV. The UAVs are simulated as 2D point mass objects with decoupled linear dynamics along the x-y axis.

[2]https://github.com/luizgiacomossi/Simulation_Wingman/tree/loyal_attacking

B. Collision Avoidance

As a primary assumption, UAVs must not collide with each other. In potential fields method, obstacles exert repulsive forces ($\overrightarrow{F}_r$) on the agent to avoid collision. An implementation of this method is by using the bivariate function [13], given by

$$f(x,y) = e^{-\alpha(x-x_c)^2-\gamma(y-y_c)^2}, \tag{2}$$

where $[x_c, y_c]^T$ is the center of the field and $[x,y]^T$ is a position on which the potential field has an influence. The x and y partial derivatives of (2) create a force field that is used for the collision avoidance among the UAVs, described by

$$\begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} -2\alpha f(x,y) & 0 \\ 0 & -2\gamma f(x,y) \end{bmatrix} \begin{bmatrix} (x-x_c) \\ (y-y_c) \end{bmatrix}. \tag{3}$$

The advantage of the bivariate function is its easy tune by constants α and γ, along the x and y axis, respectively. In the simulator, $\alpha = \gamma$ in order to generate a symmetrical field.

C. Graphical Interface

A running simulation is seen in Fig. 2. The scenario region is an open area of 17,100 m^2, divided into cells (9 x 19), where each square is equivalent to 100 m^2 (10 x 10 m). Drones are assumed to fly at altitudes of approximately 40 m.

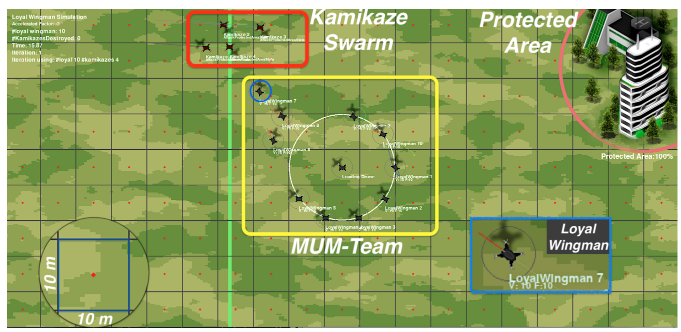

Fig. 2. Simulator view, we see the MUM-T being attacked by a kamikaze swarm. Protected area is seen on upper-right corner. A LW is seen up-close.

The graphical interface is composed of three main elements as seen in Fig. 2. The green line is used to delimit the area were the kamikaze drones will appear. The simulator also presents data about an iteration in the upper left corner, such as the number of drones on both teams, number of kamikazes destroyed by the MUM-T, execution time, iterations executed and number of LW remaining.

The user can control the position of the leading drone by point-and-click actions and the LW will follow in formation. The kamikaze and the LW drones are autonomous and the user has no control over them. The simulation run-time is adjustable in order to facilitate parameter optimization.

V. Decision Making Development

In this section, behavior development for kamikaze and the LW agents is discussed. The approach to behavior design is empirical and uses hand-coded heuristic rules.

A. Decision-Making for a Kamikaze UAV

The behaviors for the kamikaze UAVs are shown in Tab. II.

TABLE II
BEHAVIOR DESCRIPTION FOR A KAMIKAZE DRONE.

Behavior	Description
Wait State	Default behavior, a target is randomly selected.
Attack Leader	The kamikaze selects the Leading Drone as target.
Attack Protected Area (PA)	The kamikaze selects the Protected Area as target.
Attack Loyal Wingman (LW)	The kamikaze selects the closest Loyal Wingman as target.

Fig. 3 introduces the FSM employed for the autonomous control of the kamikaze drones. The initial state for a kamikaze is the *WaitState*, he waits for t_1 seconds ($t_1 = 1\,[s]$ for first wave of kamikazes and $t_1 = 0\,[s]$ for the next waves) and then select a target to attack based on the probabilities p_1, p_2 and p_3 (equal probabilities were used). Once a target is selected, the kamikaze will remain in this state until it is destroyed. The exception is *Attack Loyal Wingman* that targets the closest LW every $t_2 = 1\,[s]$, to avoid frequent target switching. Once there are no LW left, the machine transits to attack the leader or the protected area (50% probability each).

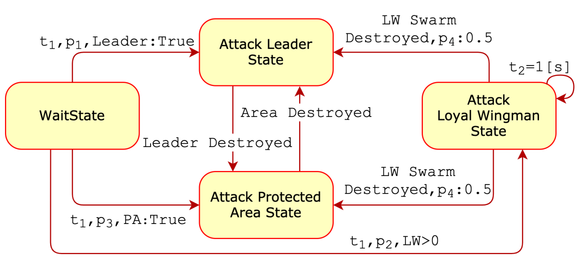

Fig. 3. Finite-state machine (FSM) for a Kamikaze agent.

B. Decision-Making for a Loyal Wingman UAV

Table III describes the possible behaviors for a LW agent.

TABLE III
BEHAVIOR DESCRIPTION FOR A LOYAL WINGMAN UAV

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

Next, we offer the agent engagement rules using these behaviors.

1) Rules of Engagement:

- By default, the LW agents execute the *Go To Formation* behavior, i.e. they keep a flying formation surrounding the leading drone while there are no threats identified;
- When a kamikaze approaches the weapons range, the LW selects a neutralization method. The selected method may be the vaporizer gun (short-range), the freezing gun (mid-range) or by sacrifice as last resort (no ammo left).
- A LW agent selects the closest kamikaze within range of his freezing gun and try to freeze him. The freezing gun must be available;
- Whenever a kamikaze drone crosses the engagement range (`distance_engagement`), the LW enters the *Chase Threat* behavior (the vaporizer gun must be available). In this case, it leaves the formation to pursue the threat and tries to neutralize it using its short-range vaporizer gun;
- A LW agent selects the closest kamikaze within range of its vaporizer gun to eliminate it. The vaporizer gun must be available;
- The LW rejoins the formation to protect the leader whenever a pursued threat is destroyed or is out of range;
- A weapon is considered available if there is still ammo and if it is not cooling down.

Thus, based on the rules of engagement and behaviors, we developed the behavior tree for a LW agent, shown in Fig.4.

VI. EXPERIMENTS AND RESULTS

In this section, we describe the experiments, the metrics for evaluation and the results. The simulations used 10 LW and 4 kamikaze drones, once one kamikaze is destroyed another is created, maintaining a steady stream of attack. The simulation set-up parameters are present in Table IV.

TABLE IV
WEAPONS PARAMETERS

Weapon	Parameter	Value
Vaporizer Gun	vaporizer_cooldown	1 [s]
	cartridges_vaporizing	10
	vaporizer_hit_prob	95%
	vaporizer_range	10 [m]
Freezing Gun	freezing_cooldown	1 [s]
	cartridges_freezing	10
	time_frozen	5 [s]
	freezing_hit_prob	85%
	freezing_range	30 [m]

PSO	Value
num_particles	40
inertia_weight	0.7
cognitive_parameter	0.6
social_parameter	0.8
UAV Parameters	**Value**
Mass	1 [kg]
K_v	4.5
K_p	0.625
max_speed	2 [m/s]

A. Metrics

The implemented MUM-T intelligence will be investigated against the following metrics of interest:

- ST (survival time): The duration of the simulation until the leader or the protected area is destroyed.
- ED (enemies destroyed): The number of kamikazes destroyed before the game is over.
- LWS (LW survived): The number of LW agents that survived the iteration.

Based on these metrics, a fitness function was empirically developed to optimize the behavior parameters, with each iteration of the optimization process being evaluated as

$$Fitness = ED^2 + 10ST - 1000LWS. \qquad (4)$$

The idea of (4) is to reward a bigger number of kamikazes destroyed and a longer duration time. There is also a penalty if there are LW drones left when the iteration ends, as the intelligence is not being effective.

B. Experiments

To assess the effectiveness of intelligence designed for the LW agents, we executed two experiments. The objective is to analyze the proposed BT, the parameters involved, see Fig.5, as well as analyze its effectiveness.

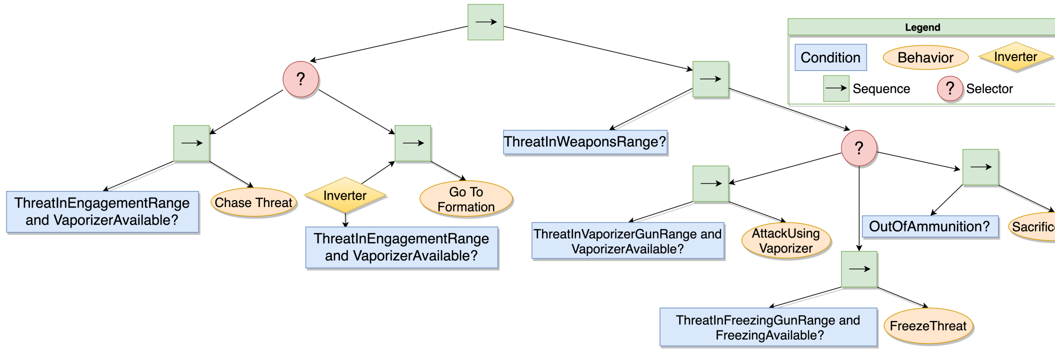

Fig. 4. Behavior tree for a loyal wingman (LW) agent.

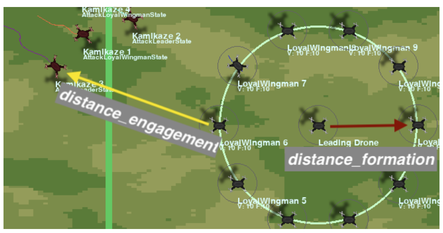

Fig. 5. Parameters description.

1) First Experiment: Test the effectiveness of the LW BT architecture and manually tune the parameters seen in Fig.5

2) Second Experiment: Optimize the LW fitness using "Particle Swarm Optimization" (PSO) algorithm [14] for maximization of the fitness function (4). The BT parameters to be optimized are seen in Fig. 5. The PSO is set using the parameters in Table. IV for 60 generations.

C. Results

The results and the parameters obtained in the experiments are shown in Table V. Using these parameters, 500 iterations were performed to obtain the statistical values of mean and standard deviation (std) for each metric.

TABLE V
EXPERIMENTAL RESULTS.

Exp.		Metrics			Parameters [m]	
		ED	ST [s]	LWS		
1	*Mean*	99.45	131.8	1.73	`distance_engagement`	36.06
	Std	8.46	13.56	1.12	`distance_formation`	13.00
2	*Mean*	102.1	167.6	1.37	`distance_engagement`	30.66
	Std	7.68	13.52	0.87	`distance_formation`	10.01

VII. DISCUSSION

First, assuming a reference to analyze the results the fact that the 10 LW used have only 100 ammunition for threat elimination (10 for each LW in vaporizer gun), and that this weapon has a 95% hit rate, *i.e.* at best we would get 110 kills – no drone misses a shot and can eliminate one threat by sacrifice when out of ammo – and a long-term average of 105 eliminations (5 misses and 10 eliminations by sacrifice).

From this, we observe the results in Tab. V. In the first experiment, the agents cooperatively eliminated an average of 99.45 kamikazes (90.4% of total threats) with a std of 8.46 kamikazes. This result indicates that the LW agents using the developed BT (Fig. 4), even with parameters found empirically, are effective as an autonomous defense system.

For the second experiment, both parameters were optimized, in order to obtain the highest defense effectiveness. The agents cooperatively eliminated an average of 102.1 kamikazes (92.8 % of the total threats), eliminating about 2.7 more enemies per iteration than in the first experiment with a smaller standard deviation. This experiment also resulted in a considerably better survival time (27.2% longer) and resulting in 20.8% less LW survivors at the end, which can indicate a better LW effectiveness.

We observed that the usage of parameter optimization resulted in a considerable improvement, with a smaller standard deviation in all indicators, contributing to a greater effectiveness of the decision-making, with results close to the optimal average of 105 threats eliminated.

A. Qualitative Analysis

The results described can also be seen on video [3]. In Fig. 6 we see how a group of LWs manage to cooperate to effectively eliminate threats. As soon as the kamikaze swarm cross the engagement limit, three LW leave the formation to neutralize the threats (*a*). In sequence, the LW group use their freezing guns to slow down the threats (*b*). In (*c*), the group destroys the threats, and all LW return to their positions in formation.

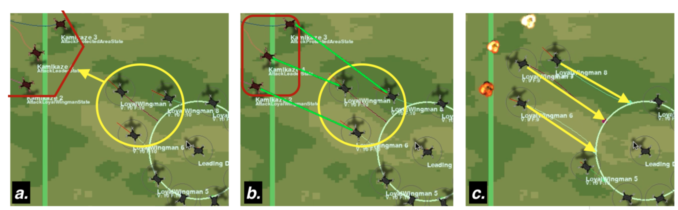

Fig. 6. Group attacking – cooperative work to neutralize threats.

Now, in Fig.7 we see a LW sacrificing itself to preserve the MUM-T leader. In (*a*) the yellow LW is seen firing his vaporization weapon and missing, so he runs away from the threat (yellow arrow). The blue LW also identifies the threat, however it has no more ammo, so it decides to execute its *SacrificeAttack* behavior.

Then in Fig.8, we have the continuation of the events after the LW sacrifice, where we observe two LW disabling a threat cooperatively. We see that the yellow and orange LWs run away after missing their targets (*a*). During the escape, the

[3]https://www.youtube.com/watch?v=bWk9x7zUZuM

Fig. 7. Sacrifice attack to preserve MUM-T as last resort tool.

yellow LW identifies the kamikaze chasing the orange LW (*b*). The orange LW succeeds in freezing the threat and then the yellow one eliminates it (*c*).

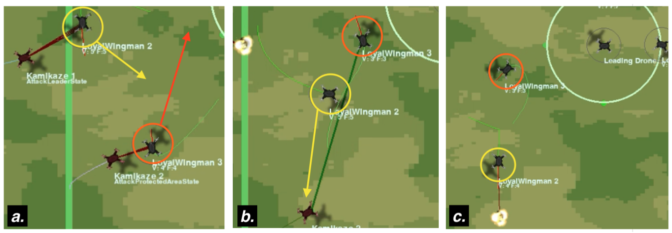

Fig. 8. Cooperative work to neutralize threat.

Lastly, in Fig.9 it is interesting to note the emergence of group responsibilities according to the kamikaze objective. The LWs marked in yellow are often responsible for forward attacks, superior lateral and to the protected area. See (a) how 3 LW leave the formation to protect the *Protected Area*.

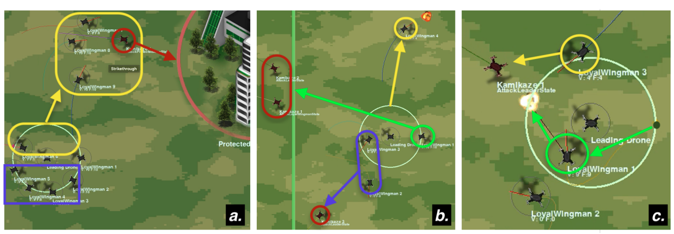

Fig. 9. Loyal Wingman specializations according to regions in formation

The LW marked in blue are responsible for attacks coming from the bottom, note in (*b*) how they end up leaving the formation when a threat appears, leaving the leader unprotected. For cases like this, the green LW stands out, it specializes in the protection of the leader (*b* and *c*). In (*c*), the yellow LW is returning to formation after a threat elimination and heading to neutralize the other kamikaze attacking the leader.

VIII. Conclusions and Future Work

In this paper, decision-making methods were employed to allow applications of autonomous drones as *Loyal Wingman* agents in a MUM-T, focusing on FSMs and BTs. Both the intelligence for LW agents and for kamikaze UAVs have been developed. Furthermore, we applied such concepts in a simulator developed for this paper. In addition, we proposed a novel defense scenario where there is a MUM-T composed of a user-controlled UAV and a swarm of completely autonomous LW UAVs. In this approach, the UAVs cooperate to engage and deactivate a swarm of kamikaze drones that are attacking both a protected area and the MUM-T. In addition, an optimization algorithm was used to find the optimal parameters for the behavior tree developed.

The results in the simulated scenario show promising signs that the developed decision-making for the LW is effective, with the elimination of about 93% of the threats after optimization. It was also noted the importance of the formation, where each drone has defense specifications. It was interesting to note that the approach using decentralized decision-making, with simple engagement rules, resulted in the emergence of cooperative team-work for threats neutralization, with the LW drones developing specialized defense behaviors, e.g. one LW exhibits a specialized behavior to defend the leader. Future research could examine how the formation affects these specializations.

Further studies should investigate possible formation optimizations and specializations according to alternative types of kamikazes. Consensus algorithms may be investigated for better use of LW, in the delegation of responsibilities and elimination of redundant attacks. Future work on the evolution of more complex behaviors, possibly using reinforcement learning, should also be carried out and analyzed. Tests with realistic physics for further evaluation should also be performed.

Acknowledgment

Luiz Giacomossi Jr. acknowledges Embraer for his scholarship.

References

[1] H. Simon, *The Science of Management Decision*. Prentice-Hall, 1960.
[2] APL-Team, "The cooperative engagement capability," Johns Hopkins, Tech. Rep. 4, 1995.
[3] E. Sepulveda and H. Smith, *Conceptual design of a fifth generation unmanned strike fighter*. AIAA Scitech 2019 Forum, 2019. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0811
[4] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Transactions on Automation Science and Engineering*, 2016.
[5] M. Buckland, *Programming Game AI by Example*. Burlington, Massachusetts, USA: Jones & Bartlett Publishers, September 2004.
[6] M. Colledanchise and P. Ogren, "Behavior trees in robotics and ai: An introduction," *CRC Press*, 07 2018.
[7] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Massachusetts, USA: The MIT Press, 2011.
[8] T. P. Baldão, M. R. O. A. Maximo, and C. Cesar, "Decision-making for 5x5 very small size soccer teams," *2020 Latin American Robotics Symposium (LARS)*, 2020.
[9] P. Ogren, "Increasing modularity of uav control systems using computer game behavior trees," *AIAA Guidance, Navigation, and Control Conference*, pp. AIAA 2012–4458, August 2012.
[10] D. C. Melo, C. H. Q. Forster, and M. R. O. A. Máximo, "Learning when to kick through deep neural networks," in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, 2019, pp. 43–48.
[11] M. R. Maximo, E. L. Colombini, and C. H. Ribeiro, "Stable and fast model-free walk with arms movement for humanoid robots," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, 2017.
[12] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "Defining "gamification"," in *From Game Design Elements to Gamefulness*, ser. MindTrek '11. New York, NY, USA: Association for Computing Machinery, 2011.
[13] L. Barnes, "A potential field based formation control methodology for robot swarms," *University of South Florida*, 2018.
[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *ICNN'95 - Inter. Conf. on Neural Networks*, vol. 4, 1995, pp. 1942–1948.