2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Cooperative protection control for UCAV swarms in hostile environments

Stanislas de Charentenay, Alexandre Reiffers-Masson and Gilles Coppin

Abstract— **Swarms of Unmanned Combat Aerial Vehicles (UCAVs) are efficient in various tasks. However, they evolve in hostile environments with risks of destruction during their flight. To mitigate this risk, it is known that cooperative behaviour can be used to enhance the protection within the swarm. The goal of this paper is to design efficient algorithms to guide the overall swarm to a given target while minimizing the risk of destruction of the member of the swarm. First, a new model, based on a controlled Markov chain, is derived to capture this cooperative swarm effect on the destruction threat of each member of the swarm. Then, an algorithm combining path planning to guide the overall swarm and local individual control to optimize the formation is suggested to help a swarm to reach a target before the destruction of all UCAVs. We evaluate our approach using numerical experiments.**

I. INTRODUCTION

Swarms of unmanned aerial vehicles (UAVs) have been studied for several years as they have a wide range of applications, such as search and rescue [1], structural and materials analysis [2], among others. They are particularly useful for military operations as they can complete complex tasks and are highly adaptable to changes in the environment. Moreover, in military operations, members of the swarm evolve in a hostile environment that will hinder the successful completion of the task. Working in hostile environments brings several challenges such as handling physical obstacles [3], communication jamming [4] or the destruction by ground-based air defense [5].

As a swarm, Unmanned Combat Autonomous Vehicles (UCAVs) can help each other throughout the mission with cooperative behaviours that increase their chances of success. In the case of a swarm at risk of destruction, defense radar saturation can be an efficient protection strategy. Indeed, it is difficult to differentiate the members of a swarm if they fly in tight formations (see references in [6], section 2.C).

In previous works, proposed solutions either aim to optimize the swarm formation or find the optimal swarm trajectories in hostile environments. We aim to solve both problems at the same time.

Our paper introduces a new model, approach, and algorithm for the design of controls of UCAVs in hostile environments and at risk of destruction. Our contribution is threefold:

Controlled Markov chain model: Our first contribution is a model based on a discrete time-controlled Markov chain with continuous and discrete states for UCAVs swarm control under attacks. A novelty of our model is that the probability of destruction of a UCAV depends not only on its position but also on the positions of other members of the swarm.

Trajectory design: We define a *reference trajectory* to guide the whole swarm. The reference trajectory is based on the optimization of a single trajectory that is further used to control the trajectories of the members of the swarm. The optimisation problem is a non-linear and nonconvex one.

Adaptive swarm formation: we derive a general approach for adaptive optimization of swarm formation where at every time instant, alive UCAVs will adjust their speed to update *their formation* to minimize a given safety criterion of the swarm.

The remainder of this paper is organized as follows. In the upcoming section, we present background and related work. Then, Section II introduces the proposed model, Section III contains the design of the guide trajectory and our adaptive control algorithms are derived in section IV. Section V reports the results of our experiments and Section VI concludes.

A. Related works

There exists a lot of work on the control of a single UAV evolving in hostile environments. In [5], the authors studied stealth problems to avoid the destruction of one UCAV from radar-guided missiles. In a more general case, with static obstacles and threat zones, a path planning algorithm for one UAV using Mixed Integer Programming (MILP) is proposed in [7]. In [3], the problem of online path planning in a partially unknown environment is addressed with a multi-step algorithm to overcome static threats such as radars or obstacles, and dynamic threats unknown to the UCAV. Moreover, optimal closed-loop controls to reach a target under various constraints have been extensively studied. In [8], classic control optimization algorithms are used to intercept a target with the optimal angle. Reinforcement Learning (RL) algorithms are also particularly efficient in this case as shown in [9] and [10]. However, all these references do not tackle the control of multiple agents. One popular UCAV swarm control problem is cooperative homing. Cooperative homing aims to synchronize several missiles to track and reach a target at the same time. In this case, the difficulty is to overcome the imperfect communication within the swarm and the imperfect observation of the target, which calls for multi-agent solutions. Solutions are usually found in the form of guidance law which can be deduced from traditional optimisation techniques like in papers [11]- [12] or from machine learning techniques [13].

Autonomous swarms can be difficult to manage, and the optimization of swarm formation is of interest. In [14], we see an example of flocking control with collision avoidance.

In reference [15], we see such a control applied to escape attacks on the swarm by avoiding a moving enemy. In reference [16], we see another interesting example of swarm control to reach a destination in a hostile environment. This paper suggests a combination of path planning with several control laws that will optimize the swarm formation along the way. However, the goal of this paper the formation is to keep a robust communication topology and there is no impact of the formation on the risk of destruction.

When looking to reach multiple targets in a hostile environment using a UCAV swarm, a well-studied problem is target allocation. Like in [17] and [18], the proposed solution is often to solve a path planning problem on each target for the swarm members and to choose the right target allocation in a second time. In [19], multi-agent RL is used to find an optimal policy that solves simultaneously the path planning and allocation problem. However, in these papers, the trajectories are computed at an individual level without taking into account the influence of the swarm.

In brief, it is always assumed that the risk of destruction for each swarm member is independent and not impacted by the rest of the swarm. However, this is not the case in most attack threat scenarios as the presence of other members can either interfere with the information collected by the attacker on the trajectory or have an impact on the strategy of this attacker. An example of such an effect used as an illustration in this paper is how close neighbors can play on the radar resolution to obstruct and confuse surface-to-air guided weapons. In this article, we want to integrate to our model these swarm interference on the destruction risk, in order to exploit them in our optimal control.

II. Model

In this paper, we control a swarm of n UCAVs in a 2 dimension space and aim to reach a target zone $\mathcal{Z} \subset \mathbb{R}^2$. The members of the swarm run the risk of being destroyed throughout their mission.

A. State and action spaces

1) Swarm states: For $i \in [\![1,n]\!]$ and $t \in \mathbb{N}$, we denote by $X_i(t) \in \mathbb{R}^2$ the position of UCAV i at instant t and the associated vector of the swarm the vector $\mathbf{X}(t) = [X_i(t)]_{i\in[\![1,n]\!]}$.

For $i \in [\![1,n]\!]$, we also define the random variable $D_i(t) \in \{0,1\}$ which indicates whether or not UCAV i is destroyed at time t. More precisely, $D_i(t) = 1$ when UCAV i is destroyed and $D_i(t) = 0$ when it is not. We call destruction state the vector $\mathbf{D}(t) = [D_i(t)]_{i\in[\![1,n]\!]}$.

2) Swarm actions: For $i \in [\![1,n]\!]$, we define $V_i(t) \in [-V_{max}, V_{max}]^2$ the controlled speed of UCAV i at time t, bounded by $V_{max} > 0$. We call $\mathbf{V}(t) = [V_i(t)]_{i\in[\![1,n]\!]}$ the action taken by the swarm at time t. We will denote the overall action space $\mathcal{V} = [-V_{max}, V_{max}]^{2^n}$.

B. Dynamic

We first study the evolution of the destruction state of the swarm. For $i \in [\![1,n]\!]$, we know that if UCAV i is destroyed, it will stay in this state until the end, i.e. for $t \in \mathbb{N}$, $\mathbb{P}(D_i(t+1) = 1 \mid D_i(t) = 1) = 1$. In the case where it is still functional at time t, we suppose that the probability of being destroyed between instants t and $t+1$ is known and depends on the position and destruction state of the swarm. As such, for $i \in [\![1,n]\!]$ and states $\mathbf{x} \in \mathbb{R}^{2^n}$ and $\mathbf{d} \in \{0,1\}^n$ with $d_i = 0$, we introduce the destruction rate function $\lambda_i : \mathbb{R}^2 \times \{0,1\} \to \mathbb{R}_+$ such that we get survival probability of UCAV i:

$$\mathbb{P}(D_i(t+1) = 0 \mid \mathbf{D}(t) = \mathbf{d}, \mathbf{X}(t) = \mathbf{x}) = \exp(-\lambda_i(\mathbf{x}, \mathbf{d}))$$

More precisely, we could write this destruction rate as $\lambda(x_i, d_i, \mathbf{x}_{-i}, \mathbf{d}_{-i})$ with (x_i, d_i) the destruction and position of UCAV i and $(\mathbf{x}_{-i}, \mathbf{d}_{-i})$ the positions and destruction states of all other members respectively. We will discuss at the end of the section why such an assumption is reasonable.

From these expressions, we can deduce for any $t \in \mathbb{N}$, the probability $q(\mathbf{d}' \mid \mathbf{x}, \mathbf{d}) = \mathbb{P}(\mathbf{D}(t+1) = \mathbf{d}' \mid \mathbf{D}(t) = \mathbf{d}, \mathbf{X}(t) = \mathbf{x})$ to transition to destruction state $\mathbf{d}' \in \{0,1\}^n$ knowing states $(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^{2^n} \times \{0,1\}^n$ through the equation:

$$q(\mathbf{d}' \mid \mathbf{x}, \mathbf{d}) = \prod_i \mathbb{P}\Big(D_i(t+1) = d'_i \mid \mathbf{D}(t) = \mathbf{d}, \mathbf{X}(t) = \mathbf{x}\Big) \quad (1)$$

The speed control is the discrete derivative of the position, so for $t \in \mathbb{N}$, $\mathbf{X}(t+1) - \mathbf{X}(t) = \mathbf{V}(t)$.

To summarize, we have the following equations describing the evolution of the different states at time $t \in \mathbb{N}$:

$$\begin{cases} \mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{V}(t), \\ \mathbf{D}(t+1) \sim q(\cdot | \mathbf{X}(t) + \mathbf{V}(t), \mathbf{D}(t)). \end{cases}$$

C. Control model

In the framework described below, the control of the swarm is characterized by the choice of an action $\mathbf{V}(t)$ for each instant t. The choice of this action $\mathbf{V}(t) = \pi(\mathbf{x}, \mathbf{d})$ will be based on the current observed states of the swarm $\mathbf{x} \in \mathbb{R}^{2^n}$.

For each time-step $t \in \mathbb{N}$:

1) We observe the positions $\mathbf{X}(t)$ and destruction state $\mathbf{D}(t)$ of the swarm;
2) Once the states are known, we choose the swarm speed $\mathbf{V}(t) \in \mathbb{R}^2$ according to the chosen control such that $\mathbf{V}(t) = \pi(\mathbf{X}(t), \mathbf{D}(t))$;
3) The swarm positions are adapted with $\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{V}(t)$;
4) The next destruction state is sampled with the destruction rate according to $\mathbf{X}(t+1)$ and $\mathbf{D}(t)$: $\mathbf{D}(t+1) \sim q(\mathbf{X}(t+1), \mathbf{D}(t))$.

III. Trajectory Planning

The first part of our approach aims to find a trajectory that guides the overall swarm to ensure a minimal safety on the overall path. In this paper, we choose our guide trajectory to be the safest path for a single UCAV to reach the target, starting from an initial position $x_0 \in \mathbb{R}^2$. This choice will be explained at the end of the section.

In this problem, we suppose that the UCAV must reach the target before a horizon $t_f \in \mathbb{N}$. As such, our path is defined by a deterministic succession of speed actions $\mathbf{v} = \{\mathbf{v}(t)\}_{t \in [\![0, t_f]\!]}$ and for $t > t_f$, we will consider that the UCAV stops moving with actions $\mathbf{v}(t) = 0$. As we only have one UCAV we use the simplified notation for destruction rate: $\overline{\lambda}(x) = \lambda(x, 0, \mathbf{0}, \mathbf{1})$.

A. Path evaluation of a single UCAV

For a given initial position $X^{(r)}(0) \in \mathbb{R}^2$ and a given succession of chosen speeds $V^{(r)} := \{V^{(r)}(t)\}_{t \in \mathbb{N}}$, the evolution of the guide trajectory is given by:

$$X^{(r)}(t+1) = X^{(r)}(t) + V^{(r)}(t).$$

For $t \in \mathbb{N}$ we have transition probability:

$$\mathbb{P}(\mathbf{D}(t+1) = 0 \mid \mathbf{D}(t) = 0) = \exp(-\overline{\lambda}(X^{(r)}(t+1))).$$

As such, for a given $V^{(r)}$ and $t \in \mathbb{N}$, and considering that $\mathbb{P}(D(0) = 0) = 1$ we can compute the closed-form expression for the probability to survive at time t:

$$\begin{aligned} \mathbb{P}(\mathbf{D}(t) = 0) &= \prod_{t'=1}^{t} \mathbb{P}(\mathbf{D}(t'+1) = 0 \mid \mathbf{D}(t') = 0) \\ &= \prod_{t'=1}^{t} \exp(-\overline{\lambda}(X^{(r)}(t')). \end{aligned}$$

from the guide trajectory, we can calculate the hitting time $\tau(\mathbf{V}^{(r)}) \in \mathbb{N} \cup \{+\infty\}$ at which we reach the target, given by:

$$\tau(V^{(r)}) = \sup\{t \in \mathbb{N} \cup \{+\infty\} \mid X^{(r)}(t) \notin \mathcal{Z}\}. \quad (2)$$

The deterministic trajectory gives us simple expression of the metric $\mathbb{P}(D(\tau(\mathbf{V}^{(r)}) = 0)$:

$$\mathbb{P}(\mathbf{D}(\tau(V^{(r)}) = 0) = \exp(-\sum_{t'=0}^{\tau(V^{(r)})} -1\overline{\lambda}(X^{(r)}(t')))$$

To determine the guide trajectory, we solve the optimization problem:

$$\min_{V^{(r)}} \exp\Big(-\sum_{t'=0}^{\tau(V^{(r)})-1} \overline{\lambda}(X^{(r)}(t'))\Big) + \gamma d(X^{(r)}(t_f))$$

subject to:

$$X^{(r)}(t+1) = X^{(r)}(t) + V^{(r)}(t), \quad (3)$$
$$V^{(r)}(t) \in \mathcal{V}, \ \forall t \in \mathbb{N}. \quad (4)$$

with $\gamma > 0$. We denote by V^* the argmin of the optimization problem and X^* the associated trajectory.

To solve this problem, we use a simple projected gradient algorithm (see section 3.3.2 in [20]). Note that the problem is in general non-convex and therefore the projected gradient algorithm will only converge to a local minimum. One can find the convergence rate in proposition 3.3 p. 213 in [20].

B. Discussion

Choosing the optimal path for a single UCAV as a guide trajectory is just one of many options, but it has the advantage of low computational complexity. Regarding the validity of such a choice, we know that in the studied problem, the destruction rate is composed of an environment risk induced by the position of the UCAV to the radars and a local swarm effect based on the swarm formation and the position of neighbours. We have chosen to deal with these two components separately, that is to say, to focus on the individual position of each member first, and to optimize their relative position in the formation optimization in a second time. This is why we chose our guide trajectory to be the optimal path of a single drone. Although we know each drone member will not follow the trajectory exactly, we know that UCAVs in an optimal swarm formation will be close together as we focus on the saturation effect of the swarm on the radars.

IV. Adaptive formation algorithms

A. Overall approach

In this section, we derive algorithms that optimize the safety of a swarm formation while following a predetermined trajectory. We first give an overall structure of these algorithms and will detail their differences later.

Let $\mathbf{X}^* = \{X^*(t)\}_{t \in \mathbb{N}}$ be the guide trajectory for our swarm. At each instant, the associated position $X^*(t)$ in this guide trajectory helps to constrain the actions of our swarm to ensure minimal safety for the path of each swarm member. We also add dynamic constraints by bounding the speed and acceleration for the taken actions. We will name the constraint space $G(\mathbf{X}(t), \mathbf{D}, X^*(t)) \subset \mathcal{V}$ and the corresponding projection operator Γ, so that $\Gamma(\mathbf{V}(t), \mathbf{X}(t), \mathbf{D}, X^*(t))$ is the projection of action $\mathbf{V}(t)$ given $\mathbf{X}, \mathbf{D}(t)$ and $X^*(t)$ at time $t \in \mathbb{N}$.

In the constraint space given by the trajectory, we optimize the swarm actions to minimize a cost $c(\mathbf{X}(t), \mathbf{D}, \mathbf{V}(t))$ dependent on the current state and taken action. Here the cost should be easy to calculate and have a low complexity in terms of swarm size n so that the optimization can be done efficiently with a projected gradient descent at each time step. We have the following overall algorithm:

For each time-step $t = 0, \ldots, t_f - 1$:

1) We observe the positions $\mathbf{X}(t)$ and destruction state $D(t)$ of the swarm, as well as the guiding trajectory position $X^*(t)$;
2) Initialize $\mathbf{V}^0(t) = \mathbf{0}$
3) *For each iteration $l = 0, \ldots, l_f - 1$ until convergence:*
    a) Calculate $c(\mathbf{X}(t), \mathbf{D}(t), \mathbf{V}^l(t))$
    b) $\mathbf{V}^{l+1}(t) \leftarrow \mathbf{V}^l(t) - \nabla c(\mathbf{X}(t), \mathbf{D}(t), \mathbf{V}^l(t))$
    c) $\mathbf{V}^{l+1}(t) \leftarrow \Gamma(\mathbf{V}^{l+1}(t), \mathbf{X}(t), \mathbf{D}(t), X^*(t))$
4) We take optimal action $\mathbf{V}^{l_f}(t)$.

Note at every time-step, the previous algorithm is a projected gradient algorithm on an optimisation problem which is in general non-convex. Therefore the projected gradient algorithm will only converge to a local minimum.

B. Suggested approaches

1) Planned leader trajectory with protectors: In this first approach, we choose member i as the "leader" of the swarm that will follow the guide trajectory $\mathbf{X}^*$. The actions of the rest of the swarm will focus on creating the best formation possible to protect this leader. We choose as constraint space:

$$\begin{aligned} G(\mathbf{X}(t), \mathbf{D}(t), X^*(t)) = \{&\mathbf{v} \in \mathcal{V}^n; \\ &\|\tilde{v}_i - \mathbf{V}_i(t-1)\|_2^2 < a_{max}; \forall i \in [\![1, n]\!]; \\ &v_i = X^*(t) - X_i(t)\}, \end{aligned} \tag{5}$$

and for all states $(\mathbf{x}, \mathbf{d})$ and action $\mathbf{v}$, we use the cost function:

$$c(\mathbf{x}, \mathbf{d}, \mathbf{v}) = \lambda_i(\mathbf{x} + \mathbf{v}, \mathbf{d}).$$

If the leader drone is destroyed, we can choose another leader and continue the algorithm.

To summarize, at every instant t we solve optimization problem:

$$\begin{aligned} \underset{v \in V}{minimize} \quad & \lambda_i(\mathbf{X}(t) + \mathbf{v}, \mathbf{D}(t)) \\ \text{subject to} \quad & \|v_j - V_i(t-1)\|_2^2 < a_{max}, \quad \forall j \in [\![1, n]\!], \\ & v_i = X^*(t) - X_i(t). \end{aligned} \tag{6}$$

2) Constraint trajectory: For this case, we want the distance of the members to the trajectory X^* to be smaller than $\rho > 0$ and to minimize the average number of detected members. We use the following constraint space and cost. The constraint space is equal to:

$$\begin{aligned} G(\mathbf{X}(t), \mathbf{D}(t), X^*(t)) = \{&\mathbf{v} \in \mathcal{V}; \\ &\|v_i - V_i(t-1)\|_2^2 < a_{max}, \forall i \in [\![1, n]\!]; \\ &\|X_i(t) + v_i - X^*(t)\|_2^2 < \rho, \forall i \in [\![1, n]\!], \} \end{aligned}$$

and for state $(\mathbf{x}, \mathbf{d})$ and action $\mathbf{v}$, the cost function is given by:

$$c(\mathbf{x}, \mathbf{d}, \mathbf{v}) = \sum_{i=1}^{n} (1 - d_i) \lambda_i(\mathbf{x} + \mathbf{v}, \mathbf{d})$$

To summarize, at every instant t, we solve optimization problem:

$$\begin{aligned} \underset{v \in V}{minimize} \quad & \sum_{i=1}^{n} \lambda_i(\mathbf{X}(t) + \mathbf{v}, \mathbf{D}(t)) \\ \text{subject to} \quad & \|v_i - V_i(t-1)\|_2^2 < a_{max}, \quad \forall i \in [\![1, n]\!], \\ & \|X_i(t) + v_i - X^*(t)\|_2^2 < \rho \quad \forall i \in [\![1, n]\!]. \end{aligned} \tag{7}$$

V. Numerical analysis

A. Example scenarios

In this section, we analyse the performance and efficiency of the proposed algorithms on two example scenarios with different destruction rates. For both scenarios, We have a number of 5 swarm members that aim to reach a target zone $\mathcal{Z}$, a closed ball of radius 0.5. They are attacked by ground air missiles guided by 2 radars. Target, radars, and initial positions of the swarm are placed as seen in figure 1. Hyper-parameters of the simulations are available in table V-A. Please note that they are not realistic and that our scenarios only have an illustrative purpose.

Scenario I: The first destruction rate model we use is based on the saturation of a radar with a low resolution when living swarm members are closed together. We want the destruction rate value to decrease as the distance to the radars increases. We also want a model where the presence of other members has no impact if they are far away from each other, but decreases the destruction rate to 0 if we have a high density of swarm members surrounding it. Let $\alpha_1, \alpha_2, \alpha_3 > 0$ be constants and $k \in \mathbb{N}$ the number of radars at positions $\{x_l^R\}_{l \in [\![1,k]\!]}$. We suggest the following expression for destruction rate of member $i \in [\![1, n]\!]$ for swarm states $\mathbf{x} \in \mathbb{R}^{2^n}$ and $\mathbf{d} \in \{0, 1\}^n$:

$$\lambda_i(\mathbf{x}, \mathbf{d}) = \sum_{l=1}^{k} \alpha_1 \frac{\left(1 + \alpha_2 \sum_j (1 - d_j) h(x_i, x_j)\right)^{-1}}{\|x_i - x_l^R\|_2^2} \tag{8}$$

$$h(x_i, x_j) = \frac{1}{1 + \alpha_3 \|x_i - x_j\|_2^2}. \tag{9}$$

Here the density around UCAV i is modeled by term $\sum_j (1 - d_j) h(x_i, x_j)$, which will only be impacted by the close presence of functional neighbors.

Scenario II: For the second scenario, we want to model the destruction risk induced by the collision between members. This is more realistic than the first scenario. We add a collision term to our previous destruction rate that becomes high when two members are too close to each other. With $\beta_1, \beta_2 > 0$ some constants, we get the expression for this second destruction rate:

$$\begin{aligned} \lambda_i(\mathbf{x}, \mathbf{d}) = & \sum_{l=1}^{k} \alpha_1 \frac{\left(1 + \alpha_2 \sum_j (1 - d_j) h(x_i, x_j)\right)^{-1}}{\|x_i - x_l^R\|_2^2} \\ & + \sum_{j=1}^{n} \left(\frac{\beta_1}{\|x_i - x_j\|_2}\right)^{\beta_2}. \end{aligned}$$

B. Simulations

In both cases, we will study the performances of the "leader protection" algorithm and the "swarm constraint" algorithm.

We will compare them with a naive algorithm consisting of ignoring the swarm effect and planning the optimal single UCAV trajectory for each member independently as in Fig. 4. As the collision effect is absent in the case of a single

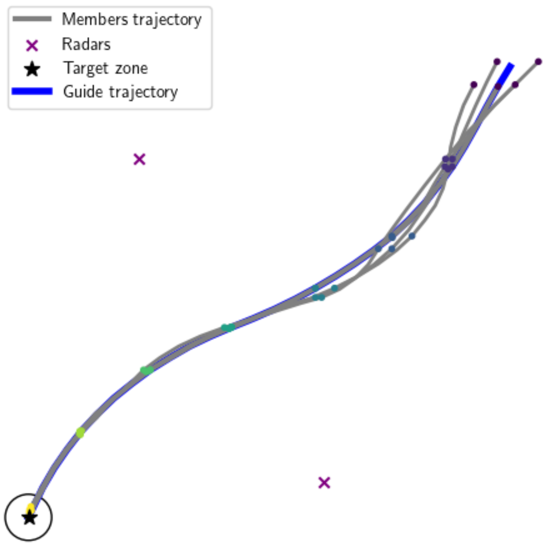

Fig. 1. Example simulation for leader protection algorithm

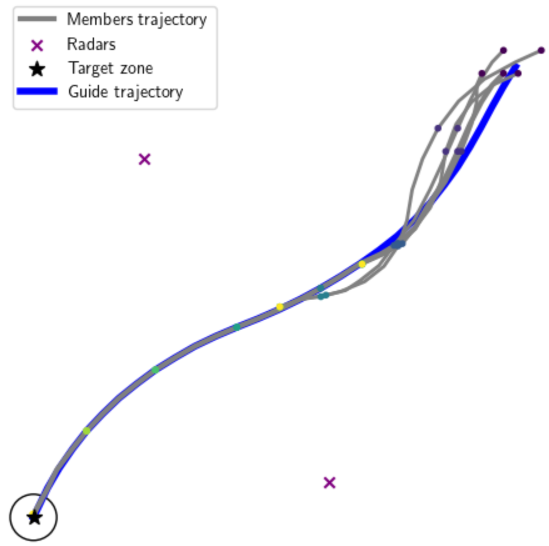

Fig. 2. Example simulation for constrained trajectory algorithm

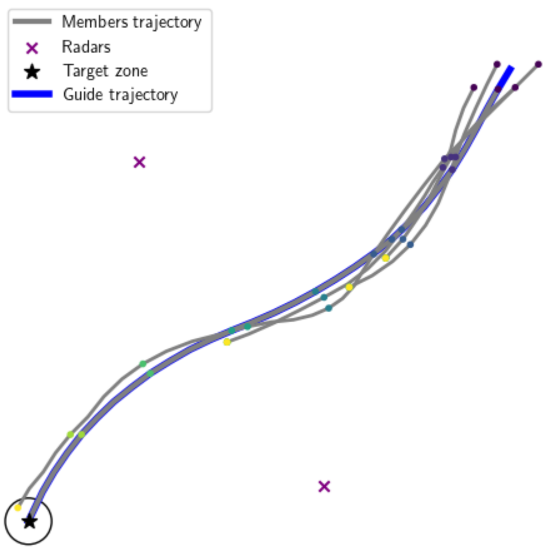

Fig. 3. Example simulation for leader protection algorithm with collision risk

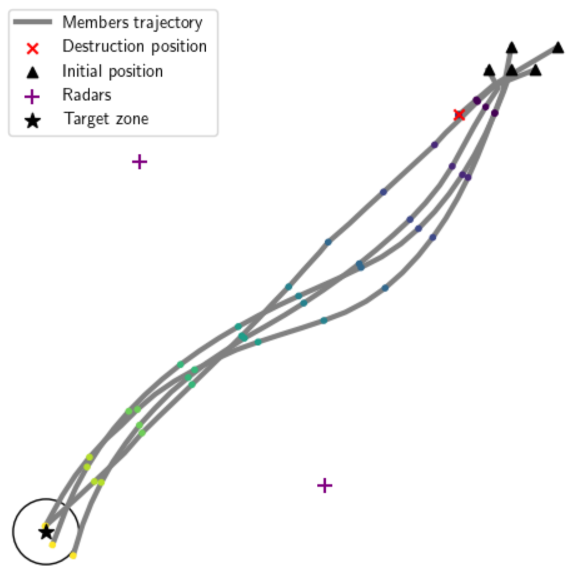

Fig. 4. Guide trajectories for each member used in naive algorithm

drone, the obtained guide trajectories are the same in both scenarios and so will be the control of the swarm by the naive approach.

To analyse the performance of each algorithm, we use a Monte-Carlo algorithm to estimate the performances of the three algorithms in the scenario I. We will make 50 simulations for each case. We show in Fig. 1 and in Fig.2 the simulation of the leader protection and the constrained trajectory respectively in the first scenario. We see that the swarm quickly reaches the optimal protection formation which is for everyone to track the leader position. As this formation does not change when members are destroyed, the death of two UCAV during the simulation does not disturb the other members and both algorithms are efficient. For the naive algorithm, as the initial positions of the swarm members are close, the guiding trajectories that they follow keep them near to each other but they are far from an optimal formation. We can see in table II that our algorithms perform much better as the optimal formation provides a better protection effect in this scenario.

The performances in the second scenario have a lower success rate as the task is more difficult. In Fig. 3, we can see that when the collision is possible, the optimal swarm formation is disrupted when the destruction state changes. However, the swarm can respond quickly and keep a high protection of the leader.

TABLE I

HYPER-PARAMETERS USED IN SIMULATIONS

Parameter	Value
Scenario parameters	
n	5
V_{max}	2
A_{max}	0.5
t_f	10
δt	0.33
Destruction rate	
α_1	4
α_2	3
α_3	4
β_1	0.2
β_2	6

Parameter	Value
Guide trajectory optim.	
learning rate	0.05
epoch nb.	6000
Formation optim.	
learning rate	0.05
iteration nb.	2000
ρ	0.05

TABLE II

MONTE-CARLO RESULTS FOR SCENARIO I

Algorithm	Success prob.
Naive	0.32
Constrained trajectory	0.64
Leader protection	0.68

VI. CONCLUSION

In this paper, we introduced a new model for controlling UCAV swarms under threat of destruction. This model can capture the influence of the swarm formation on the risk of every member of being destroyed. In this model, the optimal control to reach a static target is too complex to compute due to the high number of dimension in the state and action spaces. To overcome this, we suggest a two-step algorithm based on a global guide trajectory optimization before the beginning of the scenario and then on updating the formation at each time step by solving an optimization problem. This algorithm gives satisfactory results in the case of a tight initial formation and has the advantage of a low computational cost. Further works can include different ways to guide the swarm like a guide policy.

REFERENCES

[1] C. Sampedro, H. Bavle, J. L. Sanchez-Lopez, R. A. S. Fernández, A. Rodríguez-Ramos, M. Molina, and P. Campoy, "A flexible and dynamic mission planning architecture for uav swarm coordination," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 355–363, IEEE, 2016.

[2] J. L. Vian, A. R. Mansouri, and E. W. Saad, "System and method for inspection of structures and objects by swarm of remote unmanned vehicles," Nov. 15 2011. US Patent 8,060,270.

[3] N. Wen, L. Zhao, X. Su, and P. Ma, "Uav online path planning algorithm in a low altitude dangerous environment," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 2, pp. 173–185, 2015.

[4] C. Han, A. Liu, K. An, G. Zheng, and X. Tong, "Distributed uav deployment in hostile environment: A game-theoretic approach," *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 126–130, 2021.

[5] P. T. Kabamba, S. M. Meerkov, and F. H. Zeitz III, "Optimal path planning for unmanned combat aerial vehicles to defeat radar tracking," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 279–288, 2006.

[6] W. Khawaja, M. Ezuma, V. Semkin, F. Erden, O. Ozdemir, and I. Guvenc, "A survey on detection, tracking, and classification of aerial threats using radars and communications systems," *arXiv preprint arXiv:2211.10038*, 2022.

[7] J. J. Ruz, O. Arévalo, J. M. de la Cruz, and G. Pajares, "Using milp for uavs trajectory optimization under radar detection risk," in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pp. 957–960, IEEE, 2006.

[8] S. Xiong, M. Wei, M. Zhao, H. Xiong, W. Wang, and B. Zhou, "Hyperbolic tangent function weighted optimal intercept angle guidance law," *Aerospace Science and Technology*, vol. 78, pp. 604–619, 2018.

[9] B. Gaudet, R. Furfaro, and R. Linares, "Reinforcement learning for angle-only intercept guidance of maneuvering targets," *Aerospace Science and Technology*, vol. 99, p. 105746, 2020.

[10] S. He, H.-S. Shin, and A. Tsourdos, "Computational missile guidance: A deep reinforcement learning approach," *Journal of Aerospace Information Systems*, vol. 18, no. 8, pp. 571–582, 2021.

[11] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, "Homing guidance law for cooperative attack of multiple missiles," *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 275–280, 2010.

[12] K. Ma, H. K. Khalil, and Y. Yao, "Guidance law implementation with performance recovery using an extended high-gain observer," *Aerospace Science and Technology*, vol. 24, no. 1, pp. 177–186, 2013.

[13] X. Lan, J. Chen, Z. Zhao, and T. Zou, "Cooperative guidance of multiple missiles: A hybrid coevolutionary approach," *IEEE Transactions on Control Systems Technology*, 2023.

[14] J. Park, H. J. Kim, and S.-Y. Ha, "Cucker-smale flocking with interparticle bonding forces," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2617–2623, 2010.

[15] X. Lan, W. Xu, Z. Zhao, and G. Liu, "Autonomous control strategy of a swarm system under attack based on projected view and light transmittance," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 648–655, 2021.

[16] N. P. Borges, C. G. Ghedini, and C. H. C. Ribeiro, "Sun-stealth uav networks: A cooperative local model to mitigate uavs exposure to threats," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 4316–4323, IEEE, 2019.

[17] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, "Cooperative path planning for multiple uavs in dynamic and uncertain environments," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 3, pp. 2816–2822, IEEE, 2002.

[18] Y.-J. Eun and H. Bang, "Cooperative control of multiple ucavs for suppression of enemy air defense," in *AIAA 3rd" Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, p. 6529, 2004.

[19] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning," *IEEE access*, vol. 7, pp. 146264–146272, 2019.

[20] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.