Journal of Aerospace Information Systems, Volume 18, Issue, 10 October 2021, pp. 739-720
DOI:10.2514/1.I010961

Cooperative Planning for an Unmanned Combat Aerial Vehicle Fleet Using Reinforcement Learning

Burak Yuksek*
Cranfield University, Cranfield, England MK43 0AL, United Kingdom
Mustafa Umut Demirezen†
Digital Transformation Office, Presidency of the Republic of Turkey, 06550 Ankara, Turkey
and
Gokhan Inalhan‡ and Antonios Tsourdos§
Cranfield University, Cranfield, England MK43 0AL, United Kingdom

https://doi.org/10.2514/1.I010961

In this study, reinforcement learning (RL)-based centralized path planning is performed for an unmanned combat aerial vehicle (UCAV) fleet in a human-made hostile environment. The proposed method provides a novel approach in which closing speed and approximate time-to-go terms are used in the reward function to obtain cooperative motion while ensuring no-fly-zones (NFZs) and time-of-arrival constraints. Proximal policy optimization (PPO) algorithm is used in the training phase of the RL agent. System performance is evaluated in two different cases. In case 1, the warfare environment contains only the target area, and simultaneous arrival is desired to obtain the saturated attack effect. In case 2, the warfare environment contains NFZs in addition to the target area and the standard saturated attack and collision avoidance requirements. Particle swarm optimization (PSO)-based cooperative path planning algorithm is implemented as the baseline method, and it is compared with the proposed algorithm in terms of execution time and developed performance metrics. Monte Carlo simulation studies are performed to evaluate the system performance. According to the simulation results, the proposed system is able to generate feasible flight paths in real-time while considering the physical and operational constraints such as acceleration limits, NFZ restrictions, simultaneous arrival, and collision avoidance requirements. In that respect, the approach provides a novel and computationally efficient method for solving the large-scale cooperative path planning for UCAV fleets.

I. Introduction

IN RECENT applications of aerial attack and defense scenarios, unmanned combat aerial vehicles (UCAVs) are used to perform surveillance, reconnaissance, and neutralization of the enemy assets that are placed in human-made hostile environments. Different types of enemy defense units may be used in the warfare environment such as anti-aircraft artilleries (AAA), surface-to-air missiles (SAM), detection/tracking radars, and communication systems. These assets are selected and placed as a function of strategic importance of the defended units and geographical specifications of the defended area. By using the communication systems and defense units, a seamless air defense system can be developed to protect the ground assets. General overview of a sample warfare environment is given in Fig. 1. From the attacker fleet point of view, it is aimed to destroy the enemy assets with minimum kill probability of the fleet agents. It is also desired to complete the mission with minimum detecting and tracking probability if stealthiness is required in the operation. This can be obtained in two ways: 1) by using stealth aircraft if the flight path must pass through the enemy radar area, and 2) by generating the flight path that does not pass through the enemy radar area. The second option may be considered for minimum risk if the mission requirements and warfare environment conditions are suitable. Hence, flight path planning has a crucial importance to generate feasible and safe flight route that increases the mission success and survival probability in the warfare environment. This study focuses on the second way by developing an reinforcement learning (RL)-based cooperative centralized path planning application in which flight route is generated with minimum occupancy in the area of enemy assets while considering mission and system requirements.

Cooperation of the aerial vehicles is another important issue in the air-to-ground attack scenario. In [1], it is stated that cooperation of the autonomous UAV systems means resource sharing, information sharing, task allocation, and conflict resolution. It requires advanced sensors and long-range data link to increase the mission success and survivability of the UCAV fleet. From the survivability point of view, the cooperation is quite important to avoid possible collisions between the UCAVs. Hence, safe agent-to-agent distance should be considered while performing the flight path planning. Relative geometry data, which define distance and angle between the UCAV agents, can be used to evaluate this situation and generate a collision-free flight route. In addition, from the mission success point of view, the cooperation can be used to generate the flight path that provides simultaneous arrival in the target area. Simultaneous arrival is a critical operational concept for air-to-ground attacks to saturate the enemy air defense systems in the warfare environment. For example, if the UCAVs in the fleet infiltrate into the target area and attack to the enemy assets at the same time, the air defense system can be saturated and it cannot respond effectively against the UCAV fleet. This increases the mission success probability although it may decrease the survivability of the several UCAV agents in the fleet.

Cooperative path planning for the UCAV fleet in the warfare environment is a complex problem. As we mentioned before, there are many enemy assets that should be considered while generating the desired path. A successful cooperation is obtained by combining the survivability and mission success requirements of the fleet that are defined by the operator.

In literature, many studies are performed on cooperative path planning for the UCAV fleets. In [2], path planning for a UCAV fleet is applied by using the potential field method to suppress surface-based enemy assets such as radars, SAMs, and artilleries. In addition,

Received 15 January 2021; revision received 19 April 2021; accepted for publication 21 May 2021; published online Open Access 6 July 2021. Copyright © 2021 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Postdoctoral Research Fellow; burak.yuksek@cranfield.ac.uk. Professional Member AIAA.

†Head of Artificial Intelligence and Big Data Department; umut.demirezen@cbddo.gov.tr.

‡BAE Systems Chair, Professor of Autonomous Systems and Artificial Intelligence; inalhan@cranfield.ac.uk. Associate Fellow AIAA.

§Head of Centre for Autonomous and Cyber-physical Systems; a.tsourdos@cranfield.ac.uk.

Published by AIAA. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial License (CC:BY:NC 4.0).
The final published version (version of record) is available online at DOI:10.2514/1.I010961. Please refer to any applicable publisher terms of use.

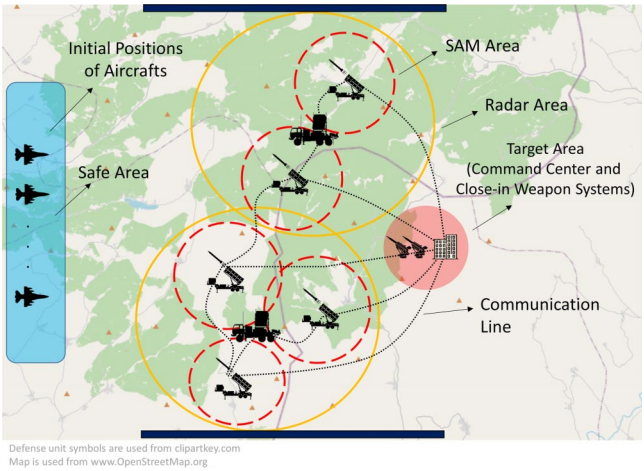


Fig. 1 General overview of a warfare environment.

Voronoi diagram is also used for the same problem, and it is compared with the performance of the proposed algorithm. Although the generated path is continuous and smooth, it requires high computation cost to solve the problem. In [3], trajectory planning problem of the UCAV fleet for a cooperative air-to-ground attack mission is formulated by integrating approximate allowable attack region model, constraints, and a multicriterion objective function. Then, virtual motion camouflage (VMC) is developed by combining differential flatness theory, Gauss pseudospectral method (GPM), and nonlinear programming to solve the cooperative trajectory optimal control problem. The performance of the proposed VMC algorithm is compared with the GPM-based direct collocation method that is developed for optimal trajectory generation. The simulation results show that the proposed method is able to generate feasible flight trajectories faster than the GPM algorithm although a small loss is observed in the optimization performance that causes suboptimal solutions.

Recent advances in computation and communication capability of the aerial vehicles have accelerated the studies on the cooperation. Application of the RL for path planning of autonomous vehicles is an emerging topic in the literature because of its ability to solve complex problems when they are formulated properly. In [4], authors developed a time-efficient navigation policy for an autonomous ground vehicle by using deep reinforcement learning (DRL). They introduced socially aware collision avoidance method with DRL and generalized it for multi-agent scenarios. The proposed algorithm is tested in a pedestrian-rich environment. In [5], a hybrid algorithm is developed that contains DRL and force-based motion planning method. It is used to solve distributed motion planning problems in dynamic and dense environments. According to the simulation results, the proposed algorithm generates up to 50% more successful scenarios than DRL method and requires up to 75% less extra time than the force-based motion planning to reach the goal. In [6], interference-aware path planning algorithm is developed for a network of cellular-connected UAV group. In this application, there is a tradeoff between the energy efficiency and wireless latency and interference. DRL algorithm based on echo state network is proposed to solve the path planning problem. Simulation results show that better wireless latency per UAV and rate per ground user are achieved when compared with the heuristic baseline method. Also, the simulation results pointed out the relationship between the optimal altitude of the UAVs, data rate requirements, and ground network density. In [7], DRL is used for distributed wildfire surveillance by using the autonomous aircrafts. In this problem, it is quite complex to maximize the forest fire coverage because of high-dimensional state space, stochastic fire propagation, imperfect sensor information, and required coordination between the aircrafts. Two DRL approaches are developed. In the first one, aircrafts are controlled by using the immediate observations from the individual aircrafts. In the second approach, wildfire state and a time history of locations that are visited by the aircrafts are used as inputs to the controller to provide collaboration between the aircrafts. According to the simulation results, the proposed approaches provide accurate tracking of the wildfire expansion and outperform the receding horizon controller. It is also stated that the approaches are scalable for different numbers of aircraft and different wildfire shapes. In [8], DRL algorithm is used to solve cooperative path planning problem for unmanned surface vehicle (USV) fleet. A leader–follower strategy is used, and a centralized coordination scheme is developed. To provide cooperation in the fleet, reward function elements related to collision avoidance and formation shape are used. However, simultaneous arrival is not considered in the path planning problem.

Multi-agent reinforcement learning (MARL) is also an emerging method to solve the multi-agent problems that contain cooperation requirements such as simultaneous arrival and collision avoidance [9–15]. In [16], deep recurrent multi-agent actor–critic framework (R-MADDPG) is developed for cooperation under partial observable situations and limited communication capability such as network bandwidth. The experiments show that the proposed R-MADDPG algorithm is able to handle with resource limitations and it enables coordination among agents in arriving simultaneously. However, aerial vehicle kinematics is not considered and obstacles are not included in the environment. In [17], distributed 4-D trajectory generation method is developed for multiple unmanned aerial vehicles (UAVs) by combining improved tau gravity (I-tau-G) guidance strategy and multi-agent Q-Learning (MAQL) algorithm. Collision avoidance and simultaneous arrival requirements are considered to provide the cooperation.

This study is a continuation of [18], in which an RL-based centralized path planning was performed for the UCAVs. A five-state survivability model, with search, detect, track, engage, and hit states, is integrated in the warfare environment. Training phase of the RL agent was performed by using proximal policy optimization (PPO) algorithm. To evaluate the effectiveness of the proposed system quantitatively, performance metrics for tracking and hit probabilities were developed and used in the Monte Carlo analysis. Simulation results indicated that the proposed algorithm was able to generate feasible flight routes while maximizing the survival probability of the UCAV fleet. However, integration of the survivability model (five states for each UCAV) into the learning process increased the observation vector size and complicated scaling of the system. Also, cooperation performance of the UCAV fleet was not investigated in [18], which is the major topic of this study.

In this paper, path planning problem for the UCAV fleet is solved by using the RL method. A centralized structure is used in which the total observation vector is fed into the single RL agent and the total action vector is generated that contains the individual control signal for the related UCAV. Unlike [18], survivability model is not integrated into the observation vector to reduce the vector size. Instead, no-fly-zones (NFZs) are defined that simulate air defense systems such as SAMs and artilleries. In addition to the studies conducted in [18], cooperation of the fleet is especially focused on here, which is obtained in twofold: Firstly, simultaneous arrival of the UCAV fleet into the target area is studied, which is a widely used method to saturate the air defense systems of the enemy. Secondly, collision avoidance is also studied to provide the fleet safety. The reward function is developed by considering these requirements. Training phase of the RL agent is performed by using the PPO algorithm. Several performance metrics are developed for NFZ avoidance, collision avoidance, and simultaneous arrival requirements to obtain a quantitative evaluation of the proposed method. By using Monte Carlo analysis, cooperation performance of the system is evaluated based on collision avoidance and simultaneous arrival capabilities of the fleet in the presence of NFZ position uncertainties and external disturbances (i.e., wind effects).

This study contributes to the literature from two aspects. First, to the authors' best knowledge, this is the first time that a feasible and tractable RL-based centralized path planning methodology is developed for UCAV fleets. For example, in comparison to typical PSO-based methods, the RL-based approach provides the real-time ability for the fleet to replan in face of dynamic and counterattacking/defending threats. Second, in contrast to current approaches, the

proposed methodology provides the capability to simultaneously consider key operational constraints such as simultaneous arrival and collision avoidance requirements while considering NFZ restrictions and system limitations such as lateral acceleration command limit of the UCAVs. For example, typical approaches such as PSO-based methods consider a much limited subset of these restrictions, and thus they have applicability to only some aspects of real-life scenarios. Considering both contribution facets, the proposed approach not only provides means for real-life applicable cooperative operation capability based on fundamental terms such as closing speed and approximated time-to-go information, but also provides a real-time approximation to a highly nonlinear and large-scale UCAV fleet optimization problem.

Remaining of the paper is organized as follows: In Sec. II, mathematical model and relative geometry used in the path planning problem are explained. In Sec. III, general structure of the RL agent is given and training algorithm is described. In Sec. IV, simulation results are given and evaluated for 1) without NFZ and 2) with NFZ constraints. In Sec. V, concluding remarks and future works are explained.

II. Background

As mentioned in the Introduction, this study aims to develop RL-based cooperative path planning for a UCAV fleet in the warfare environment. For this purpose, learning algorithm of the RL agent should be given clearly. In addition, basic definitions about the simulation environment should be made to generate a feasible observation vector. This section provides preliminary insight about the learning algorithm and simulation environment. First, the main idea of the PPO algorithm is described and mathematical expressions are given. Then, a simplified kinematic model of a UCAV is shared and assumptions are explained. Finally, relative geometry between the UCAV agents and enemy assets that is used in the observation vector is defined mathematically.

A. Proximal Policy Optimization Algorithm

PPO is an on-policy, model-free, policy-gradient learning method in actor–critic structure. It attains data efficiency and reliability of thrust region policy optimization (TRPO) while performing first-order optimization process. It has several advantages when compared with other learning algorithms. First, it is model-free and there is no need to obtain the state transition matrix of the dynamic system. Second, it is suitable for control problems with continuous state and action spaces. This is an important feature especially for control design applications of dynamic systems. Third, implementation and tuning of the algorithm are easy and require low computational cost. Finally, hyperparameters of the algorithm have robust characteristics against variety of tasks [19].

In this algorithm, the policy network (π) uses the system states (s) (i.e., observation data) and generates an action signal (a). The policy network is used to generate mean and variance of a multivariate Gaussian for the continuous action space. Then, the action signal is generated from this information. In the training phase, action signal is generated from this distribution and the mean value is used after the training phase is completed [19].

From the general point of view, it is important to create an observation and action vectors that contain useful information about the environment and feasible input signal to obtain the desired agent behavior. To gain insight about the definition of the observation and action vectors for this specific path planning problem, it is useful to give preliminary explanations about them. In this application, the observation vector contains states of the UCAVs and information about the relative geometry between the UCAVs and target area. The action vector contains bounded lateral acceleration command signals for each of the UCAV agents. Detailed definitions of the observation and action vectors are given in following sections.

In the policy gradient algorithms, the policy gradient is estimated and then gradient ascent algorithm is used. The commonly used gradient estimator algorithm is defined in Eq. (1).

$$\hat{g} = \hat{\mathbb{E}}_t\Big[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\Big] \tag{1}$$

where π_θ is policy network optimized with regard to parameter vector θ. Subscript t indicates the time step, $\hat{\mathbb{E}}_t$ is expectation, and a_t, s_t are action and state vectors, respectively. $\hat{A}_t$ is estimation of the advantage function and obtained by using an advantage estimator given in Eq. (2) [20,21].

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T) \tag{2}$$

where $V(s_t)$ is state-value function at time index $t \in [0, T]$, T is the length of the trajectory segment, r_t is reward at time step t, and γ is discount factor. General definition of the advantage estimation is given in Eq. (3), which reduces to Eq. (2) when the smoothing parameter is selected as $\lambda = 1$.

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{3}$$

where δ_t is given in Eq. (4):

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{4}$$

The gradient estimation $\hat{g}$ is obtained by differentiating the objective function L^{PG} given in Eq. (5).

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t] \tag{5}$$

where superscript PG stands for policy gradient.

In the PPO algorithm, it is proposed to maximize the clipped surrogate objective function $L^{\text{CLIP}}(\theta)$, which is given in Eq. (6).

$$L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t\Big[\min\Big(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\Big)\Big] \tag{6}$$

where $r_t(\theta)$ is probability ratio given in Eq. (7).

$$r_t(\theta) = \frac{\pi_\theta(a_t, s_t)}{\pi_{\theta_{\text{old}}}(a_t, s_t)} \tag{7}$$

To increase the efficiency of the sample, the PPO algorithm uses importance sampling in which dataset is used from the old policy $\pi_{\theta_{\text{old}}}$ and new policy π_θ. After the new policy is updated, variance of the estimation increases as a result of diverging of the old and new policies. Hence, the old policy is updated periodically to match the new policy. In addition, similar state transition functions are required for two policies. This is obtained by a clip operator for the probability ratio in the range of $[1-\varepsilon,\ 1+\varepsilon]$, which provides a first-order approach for the thrust region optimization. The minimum operator in the surrogate objective function [Eq. (6)] is used to bound the unclipped objective. By doing so, a pessimistic estimate of the policy performance is obtained [19,21].

General overview of the PPO algorithm is given in Algorithm 2.1. In each iteration, each of the N parallel actors collects data during T time steps and calculates advantage function estimations. Then, the surrogate objective L^{CLIP} is optimized with regard to parameter vector θ for K epochs and minibatch size $M < NT$ [21].

Algorithm 1 PPO algorithm, actor–critic style [21]

for iteration = 1, 2, . . . **do**
  for actor = 1, 2, . . . N **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T time steps;
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$;
  End
  Optimize L^{CLIP} w.r.t. θ, with K epochs and $M \le NT$;
  $\theta_{\text{old}} \leftarrow \theta$
End

B. UCAV Kinematic Model

Closed-loop UCAV systems can be modeled by using the reduced-order kinematic models that are appropriate for development of high-level motion planning and guidance algorithms [22,23]. They provide sufficient information about the kinematics of the aerial vehicle while using simplified equations of motion that does not increase computational cost. However, it is important to define several assumptions about the controlled dynamics of the aerial vehicle. Improper assumptions about the controlled dynamics may cause generation of infeasible trajectories and even loss of the aerial vehicles.

Another important issue while using the kinematic models is consideration of operational constraints such as acceleration, velocity, and altitude limits of the aerial vehicles. These data provide information about the physical limits of the aerial vehicles, and they are crucial in the motion planning applications to obtain dynamically feasible paths from the aerostructural and aerodynamic point of view.

In this study, the kinematic model structure is used with five assumptions to represent the simplified UCAV kinematics.

Assumption 1: The UCAVs have optimally designed low-level autopilots such as pitch and roll attitude controllers to track the given signals from the high-level systems and minimize the disturbance effects.

Assumption 2: Altitudes of the UCAVs are controlled by an altitude control system and it is fixed during the operation.

Assumption 3: Given the fact that the operational cruise velocities of combat air vehicles are relatively constant and mission optimized through the flight planning system, we assume that the ground speed of the UCAV is constant and controlled by a velocity control system.

Assumption 4: Lateral acceleration control system has sufficiently high bandwidth to track the given acceleration commands by the RL agent. Hence, the closed-loop lateral acceleration dynamics is modeled as a gain, $G(s) \cong 1$.

Assumption 5: Lateral acceleration limit of the UCAV platforms on the north–east plane is $\pm 8g$.

In the light of above-mentioned assumptions, mathematical definition of the kinematic model is given in Eq. (8).

$$\begin{aligned} \dot{p}_n(t) &= V_a \cos\psi(t) + w_n \\ \dot{p}_e(t) &= V_a \sin\psi(t) + w_e \\ \dot{\psi}(t) &= n_{lat_c}(t)/V_a \end{aligned} \tag{8}$$

where p_n, p_e are north and east position; ψ, V_a, and n_{lat_c} are heading angle, ground speed, and commanded lateral acceleration, respectively; and w_n, w_e are wind speed components in north and east directions that are used as external disturbances.

C. Relative Geometry

Before explanation of the system structure, it is important to define the relative geometry between the allied aircrafts and ground-based enemy assets. This information will be used to generate the observation vector that is directly related with the learning performance of the RL agent. The relative geometry between two allied aircrafts and one ground asset is given in Fig. 2a.

Generalized mathematical definition of the relative geometry is defined in Eqs. (9–12) to solve the problem in the whole warfare environment for the ith allied aircraft and kth enemy asset, where $i = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, n$. Here, m is total number of allied aircrafts, and n is total number of enemy assets.

$$d_{a_{iT_k}}(t) = \|\boldsymbol{p}_{a_i}(t) - \boldsymbol{p}_{T_k}\| \tag{9}$$

$$d_{a_{ia_j}}(t) = \|\boldsymbol{p}_{a_i}(t) - \boldsymbol{p}_{a_j}(t)\| \tag{10}$$

$$\varphi_{a_{iT_k}}(t) = a\tan 2\left(\frac{p_{y_{a_i}}(t) - p_{y_{T_k}}}{p_{x_{a_i}}(t) - p_{x_{T_k}}}\right) \tag{11}$$

$$\varphi_{a_{ia_j}}(t) = a\tan 2\left(\frac{p_{y_{a_i}}(t) - p_{y_{a_j}}(t)}{p_{x_{a_i}}(t) - p_{x_{a_j}}(t)}\right) \tag{12}$$

In these equations, $d_{a_{iT_k}}$ is distance between the ith UCAV and kth enemy asset, $d_{a_{ia_j}}$ is distance between the ith and jth UCAVs, $\varphi_{a_{iT_k}}$ is relative angle between the ith UCAV and kth enemy asset, and $\varphi_{a_{ia_j}}$ is relative angle between the ith and jth UCAVs. $\boldsymbol{p}_{a_i} = [p_{x_{a_i}}, p_{y_{a_i}}]^T$ and $\boldsymbol{p}_{T_k} = [p_{x_{T_k}}, p_{y_{T_k}}]^T$ are position vectors of ith UCAV and kth enemy asset.

Simultaneous arrival on the target area is one of the main purposes of this study. Hence, closing speed of the UCAVs toward the target point and approximate time-to-go data play a crucial role to evaluate the arrival time difference of the UCAVs. Geometrical and mathematical definitions are given in Fig. 2b and Eqs. (13) and (14) for the ith UCAV.

$$V_{c_i} = V_{a_i}\cos(\psi_i - \varphi_{a_{iT}}) \tag{13}$$

$$t_{\text{go}_i} = d_{a_{iT}}/V_{c_i} \tag{14}$$

where V_{c_i} is closing speed component on the line-of-sight between the ith UCAV and target, and t_{go_i} is approximated time-to-go of the ith UCAV. Difference of the approximated time-to-go of the ith and jth UCAVs is defined in Eq. (15).

$$\Delta t_{\text{go}_{ij}} = t_{\text{go}_i} - t_{\text{go}_j} \tag{15}$$

III. Approach

In an RL-based path planning application, it is quite critical to define an adequate reward function, observation vector, and action vector to obtain a feasible flight route that maximizes the mission success of the fleet agents. The observation vector and reward function are used to define the relationship between system input and

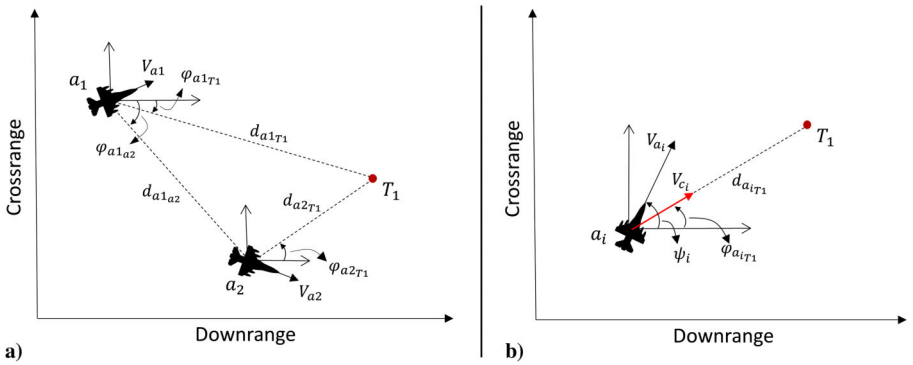


Fig. 2 Relative geometry definitions: a) relative distance and angle between two allied aircrafts and one enemy asset [18]; b) closing speed of the aircraft toward the target.

desired behavior. Hence, the main problem in RL-based path planning is to develop an adequate reward function and observation vector to generate proper action signal that is applied into the environment. In this section, a general overview of the RL-based centralized cooperative path planning will be given and the PPO training algorithm will be explained.

The proposed algorithm is developed based on the centralized scheme because of its simple structure. It is suitable for small teams and provides simple and effective solution although its scalability difficulties; i.e., computational complexity increases with the number of agents. This problem can be solved by developing decoupled (or decentralized) applications, and it is addressed as future work in the Conclusions section.

In the RL-based centralized path planning algorithm, all the data collected from the environment are provided into the RL agent, and it designs the motion for all of the UCAVs. To generate the control signal, the agent also uses the returned reward that describes the relationship between the applied input and behavior of the UCAVs. A general overview of the RL agent and its interactions with the warfare environment is given in Fig. 3.

The reward function is developed according to the mission requirements such as minimum occupation in the NFZ, collision avoidance, and simultaneous arrival into the target area. Total reward function $\boldsymbol{R}$ is defined as sum of the individual rewards of the UCAVs, R_i, and common reward R_c. Here, R_i is used as UCAV-specific reward, whereas R_c is used as common reward for the fleet such as simultaneous arrival and collision avoidance. The total reward function is given in Eq. (16).

$$\boldsymbol{R} = \sum_{i=1}^{m}(R_i) + R_c \tag{16}$$

The individual reward functions are used to evaluate the behavior of the UCAVs in the warfare environment. Many conditions are checked and the system is either rewarded or punished according to the definitions. Common reward function R_c is used to evaluate the fleet behavior instead of the behavior of the individual UCAVs. The general structure of the individual reward function for the ith UCAV and the common reward function are given in Eqs. (17) and (18), respectively.

$$R_i = w_{n_i}R_{n_i} + w_{\text{out}_i}R_{\text{out}_i} + w_{\text{stp}_i}R_{\text{stp}_i} + w_{t_i}R_{t_i} + w_{\text{act}_i}R_{\text{act}_i} \tag{17}$$

$$R_c = w_{ca}R_{ca} + w_{sa}R_{sa} + w_{\Delta t_{go}}R_{\Delta t_{go}} \tag{18}$$

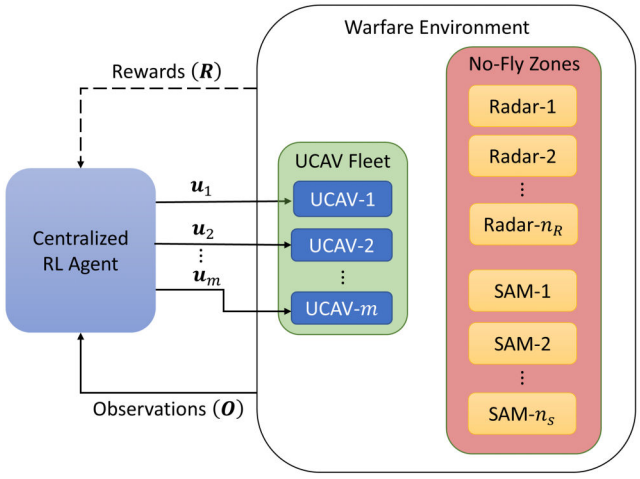


Fig. 3 General overview of the RL agent and its interaction with the warfare environment.

where $R_{n_i}, R_{\text{out}_i}, R_{\text{stp}_i}, R_{t_i}, R_{\text{act}_i}, R_{ca}, R_{sa}, R_{\Delta t_{go}} \in \mathbb{R}^+$ are rewards for being in the NFZ, being out of the warfare environment, simulation step limitation, reaching to the target area, action signal magnitude, collision avoidance, simultaneous arrival, and time-to-go bound, respectively. Individual weights w_{n_i}, w_{out_i}, w_{stp_i}, w_{t_i}, w_{act_i}, w_{ca}, w_{sa}, $w_{\Delta t_{go}} \in \mathbb{R}$ span to the multi-objective optimization Pareto-optimal frontier. In addition, these weights in the reward functions define the relative importance of each separate reward in the individual and common reward functions $R_i$andR_c, respectively. Selection of these weights is directly related to desired performance of the path planning system. For example, if it is required to minimize the collision probabilities at the cost of actuator usage, this can be achieved by increasing the collision avoidance weight w_{ca} and decreasing the actuator usage weight w_{act}. Increased collision avoidance weight increases the given punishment when the UCAVs violate the collision avoidance requirement. In addition, decreased actuator usage weight results in performing more effective collision avoidance maneuvers by generating high-amplitude acceleration commands. Combination of these two modifications on the reward function weights minimizes the collision probability of the UCAVs. As explained in this example, selection of the reward function weights is a problem-specific process and it is performed by progressive training, simulation and evaluation studies. Similar to the development of the conditional reward functions, the weight selection and its sensitivity analysis are addressed as future works and they will be investigated.

The reward function elements are defined in Eqs. (19–26).

$$R_{n_i} = \begin{cases} k_n, & \text{if } d_{a_{in_k}} < d_n \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$R_{\text{out}_i} = \begin{cases} k_{\text{out}}, & \text{if } i\text{th UCAV is out of the warfare environment} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

$$R_{\text{stp}_i} = \begin{cases} k_{\text{stp}}, & \text{if } n_{\text{stp}_i} > n_{\text{stp}_{\max}} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

$$R_{t_i} = \begin{cases} k_t, & \text{if } d_{a_{iT}} < d_T \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

$$R_{\text{act}_i} = k_{\text{act}}(n_{\text{lat}_{c_i}}/g)^2 \tag{23}$$

$$R_{ca_i} = \begin{cases} k_{ca}, & \text{if } d_{a_{ia_j}} < d_{\text{ca}} \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

$$R_{sa} = \begin{cases} k_{sa}, & \text{if } d_{a_{iT}} < d_T \text{ and } \Delta t_a < \Delta t_{a_{\max}} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

$$R_{\Delta t_{\text{go}}} = \begin{cases} k_{\Delta t_{\text{go}}}, & \text{if } \Delta t_{\text{go}} < \Delta t_{\text{go}_{\max}} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

where $d_{ca} = 50$ m is minimum allowable distance between the UCAVs, $d_T = 1500$ m is radius of the target area, $d_n = 1500$ m is radius of the NFZ area, n_{stp_i} is number of the simulation steps of the ith UCAV in each episode, $n_{\text{stp}_{\max}} = 1000$ is maximum allowable step number of simulation, $\Delta t_a \in \mathbb{R}$ is arrival time difference of the UCAVs, $\Delta t_{a_{\max}} = 2$ s is maximum allowable arrival time difference, Δt_{go} is time-to-go difference of the UCAVs calculated on each step, $\Delta t_{\text{go}_{\max}} = 2$ s is maximum allowable time-to-go difference at each simulation step, and g is gravity acceleration in m/s^2. Reward function parameters $k_n, k_{ca}, k_{\text{out}}, k_{\text{stp}}, k_t, k_{sa}, k_{\Delta t_{\text{go}}}, \in \mathbb{R}$ are defined as reward values for being in the NFZ, collision avoidance, being out of the map, simulation step limitation, being in the target area, simultaneous arrival, and time-to-go, respectively. Here, R_{act_i} is a function of k_{act} gain and $n_{\text{lat}_{c_i}}$, and it is used to minimize the

Table 1 Reward function parameters

Parameter	Value
k_n	−10
k_{out}	−300
k_{stp}	−200
k_t	100
k_{sa}	100
k_{ca}	−100
k_{act}	−0.01
$k_{\Delta t_{go}}$	−10

magnitude of the commanded lateral acceleration of the UCAVs. Conditional reward values in Eqs. (19–26) are also related with the relative importance of the each reward function. Although they can be adjusted according to the mission requirements, we fixed them and used the weights within the reward functions to define the relative importance of each reward and, consequently, to achieve the desired operational behavior. The values of these parameters are given in Table 1. These values are selected as a result of progressive training, simulation, and evaluation studies.

It is important to note that both R_{sa} and $R_{\Delta t_{\text{go}}}$ are used in the reward function to obtain the simultaneous arrival. R_{sa} is used at the end of each episode and the system should meet two requirements: a) UCAV should be in the target area, and b) difference of arrival time (Δt_a) should not exceed the given bound ($\Delta t_{a_{\max}}$). On the other hand, $R_{\Delta t_{\text{go}}}$ is used to evaluate the time-to-go difference of the UCAVs and it is calculated on each simulation step.

A feasible observation vector provides an efficient learning process and improves the agent performance by defining the input–output relationship clearly. For a warfare scenario, the observation vector should include measurable data from different types of sources such as UCAV position, orientation, and relative position between the enemy assets and UCAVs to increase the situation awareness capability of the agent. In the light of these requirements, the observation vector $\boldsymbol{O}$ contains point-mass model states, relative position, and orientation data according to the enemy assets. The observation vector of the ith aircraft is given in Eq. (27).

$$\boldsymbol{O}_{a_i} = \left[\boldsymbol{p}_{a_i}^T, \psi_{a_i}, d_{a_{iT_k}}, \varphi_{a_{iT_k}}, d_{a_{ia_j}}, \varphi_{a_{ia_j}}\right] \tag{27}$$

where $\boldsymbol{p}_{a_i} = [p_n, p_e]^T \in \mathbb{R}^2$, $\psi_{a_i} \in \mathbb{R}$ are position vector and heading angle of the ith UCAV, respectively, and $d_{a_{iT_k}}$, $\varphi_{a_{iT_k}}$, $d_{a_{ia_j}}$, $\varphi_{a_{ia_j}}$ define the relative geometry data of the UCAVs and they are given in Eqs. (9–12). The total observation vector $\boldsymbol{O}$ contains the observation data from each UCAV in the fleet and it is given in Eq. (28).

$$\boldsymbol{O} = [\boldsymbol{O}_{a_1}, \boldsymbol{O}_{a_2}, \ldots, \boldsymbol{O}_{a_m}]^T \tag{28}$$

A typical air-to-ground mission with simultaneous arrival and NFZ constraints requires two main inputs for the aircraft platforms. These inputs are velocity command ($V_{a_{c_i}}$) and lateral acceleration command ($n_{\text{lat}_{c_i}}$) for the ith aircraft. In this study, it is assumed that the UCAVs operate at constant ground speed and the proposed centralized RL agent is developed to provide lateral acceleration command for each UCAV. The action vector $\boldsymbol{u} \in \mathbb{R}^m$ is given in Eq. (29), where m is the number of UCAVs in the allied fleet. To provide a feasible control signal into the system, lateral acceleration command is bounded as $n_{\text{lat}_{c_i}} \in [-8g, +8g]$ in the training process.

$$\begin{aligned}\boldsymbol{u} &= [\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_m}]^T \\ &= \left[n_{\text{lat}_{c_1}}, n_{\text{lat}_{c_2}}, \ldots, n_{\text{lat}_{c_m}}\right]^T\end{aligned} \tag{29}$$

IV. Simulation Studies

Simulation studies are conducted on a fleet that includes 2 UCAV agents. Training parameters of the PPO algorithm are given in Table 2 for the reproducibility of the application. After performing the training phase, performance of the proposed system is evaluated by using the Monte Carlo analysis in two case studies. These cases represent different types of missions, i.e., saturated attack with and without NFZ restrictions. For these cases, initial condition intervals for UCAV-1 and UCAV-2 are given in Table 3.

A. Performance Metrics

To perform a quantitative evaluation of the proposed algorithm, it is important to define performance metrics based on critical system and/or mission specifications such as fuel consumption, control actuation usage, NFZ violation, and simultaneous arrival requirements. These metrics are also important to compare the performance of the proposed algorithm with others. As the tactical requirements such as NFZ restrictions, collision avoidance, and simultaneous arrival are primarily considered in this study, the performance metrics are developed based on these quantities.

Before explaining the performance metrics, it is useful to restate some definitions about the relative geometry between the UCAVs, NFZs, and target area just to clarify the problem. The simultaneous arrival problem is described in Fig. 4a, where $\boldsymbol{p}_{a_1}(t_k), \boldsymbol{p}_{a_i}(t_k) \in \mathbb{R}^2$ are position vectors of the 1st and ith UCAVs at $t = t_k$. The 1st UCAV reaches into the target area at $t_{f_1} = t_k + \Delta t_k$, where t_{f_1} is the arrival time. Because of the position disadvantage, the ith UCAV reaches into the target area at $t = t_{f_2}$ and arrival time difference $\Delta t_a = t_{f_1} - t_{f_2}$ is observed in the mission. As one of the main aims of this paper is to minimize Δt_a to obtain the simultaneous arrival condition, it is can be used in the simultaneous arrival performance metric $P_{sa} \in \mathbb{R}^{0+}$, which is given in Eq. (30).

$$P_{sa} = \sum_{n=1}^{n_{mc}} \frac{|\Delta t_a(n)|}{n_{mc}} \tag{30}$$

where n_{mc} is number of Monte Carlo simulations. Avoiding the NFZs is another expected capability of the proposed algorithm, which is illustrated in Fig. 4b, where $r_n \in \mathbb{R}^{0+}$ is radius of the NFZ and $d_{an}(t_k) \in \mathbb{R}^{0+}$ is distance between the UCAV and center of the NFZ at time instant $t = t_k$. If the UCAV is in the NFZ (i.e., $r_n < d_{an}(t_k)$), related elements in the position vector of the UCAV are marked as NFZ-violated points and all of the generated paths are checked for this condition. In other words, each element in the position vectors is checked whether it is in the NFZ or not. By using these data, performance metric for the NFZ violation $P_{\text{nfz}} \in [0, 1000]$ is calculated as given in Eq. (31).

Table 2 Training parameters of the PPO algorithm

Parameter	Value
Number of steps for per update	128
Number of minibatches	4
Discount factor	0.99
Factor of bias vs variance tradeoff	0.95
Learning rate	5E-5
Clip range	0.2
Number of epoch when minimizing the surrogate	4
Entropy coefficient	0

Table 3 Initial condition intervals for case 1 and case 2

	UCAV-1			UCAV-2		
Case	$p_n(0)$, km	$p_e(0)$, km	$\psi(0)$, deg	$p_n(0)$, km	$p_e(0)$, km	$\psi(0)$, deg
1	[1, 9]	[1, 2]	[−30, 30]	[10, 18]	[3, 4]	[−30, 30]
2	[6, 13]	[1, 2]	[−45, 45]	[6, 13]	[2, 3]	[−45, 45]

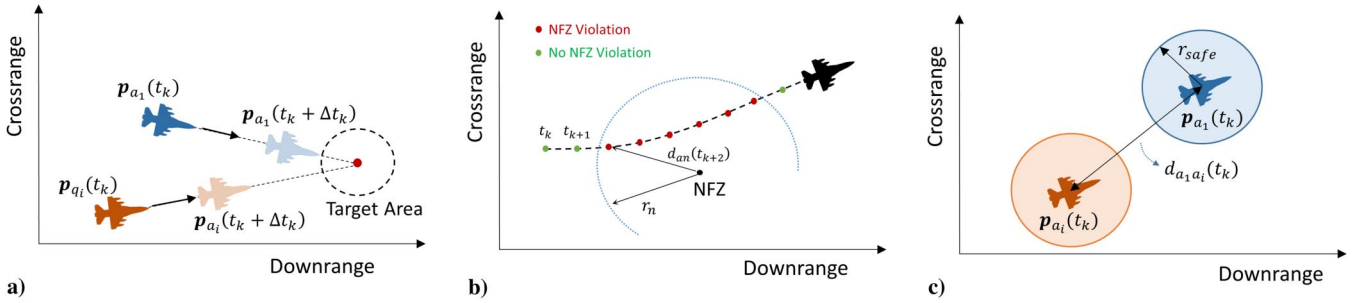


Fig. 4 Definitions for a) simultaneous arrival, b) NFZ restrictions, and c) collision avoidance.

$$P_{\text{nfz}} = 1000 \sum_{n=1}^{n_{mc}} \sum_{k=1}^{k_e} \frac{N_{\text{nfz}}(k)}{n_{mc} k_e} \tag{31}$$

where $N_{\text{nfz}} \in \mathbb{Z}^{0+}$ is total number of position samples that violate the NFZ condition in the kth episode, and $k_e \in \mathbb{Z}^{0+}$ is total number of samples in each episode.

The last capability of the proposed algorithm that should be evaluated is collision avoidance during the entire mission and it is illustrated in Fig. 4c. Here, $r_{\text{safe}} \in \mathbb{R}^{0+}$ is minimum allowable distance between the UCAVs for a safe flight and $d_{a_{1a_n}} \in \mathbb{R}^{0+}$ is actual distance between the 1st and nth UCAVs. Similar to the NFZ performance metric, the number of samples in the position vector of the UCAVs that violate the collision avoidance requirement $d_{a_{1a_n}} \geq r_{\text{safe}}$ is used in the collision avoidance performance metric $P_{ca} \in [0, 1000]$ in Eq. (32).

$$P_{ca} = 1000 \sum_{n=1}^{n_{mc}} \sum_{k=1}^{k_e} \frac{N_{ca}(k)}{n_{mc} k_e} \tag{32}$$

where $N_{ca} \in \mathbb{Z}^{0+}$ is the total number of position samples that violate the collision avoidance condition. These metrics will be used to evaluate the performance of the proposed structure and to compare it with a baseline path planning algorithm.

B. Case 1: Simultaneous Arrival Without NFZ Restrictions

In case 1, the proposed system is trained for the warfare environment that does not include any NFZ such as SAMs, artilleries, and radars. Here, it is desired to arrive to the target area simultaneously to saturate the enemy air defense system. The learning curve of case 1 is given in Fig. 5.

Position data of the UCAV fleet on the east–north plane are given in Fig. 6. Although the UCAVs start from different positions and heading conditions, they are able to reach to the target area while meeting the system and operational requirements such as acceleration command limit and simultaneous arrival.

Time history of the action signal of the RL agent, i.e., acceleration command for UCAV-1 and UCAV-2, is given in Fig. 7. To provide the clarity of the figure, time history data are plotted just for five episodes of the Monte Carlo simulation. Here, it is important to note that the

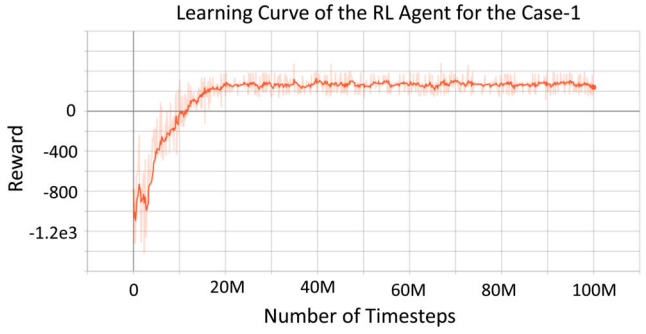


Fig. 5 Learning curve of the RL agent for case 1.

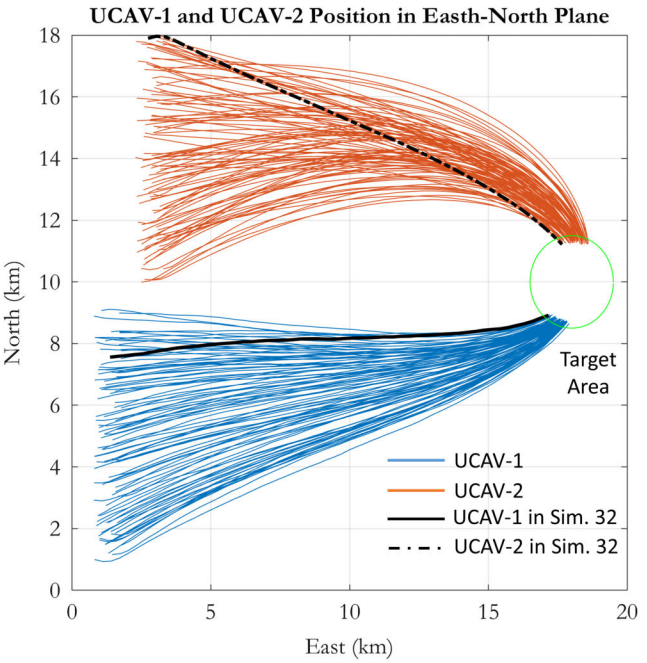


Fig. 6 UCAVs' position in east–north plane for case 1.

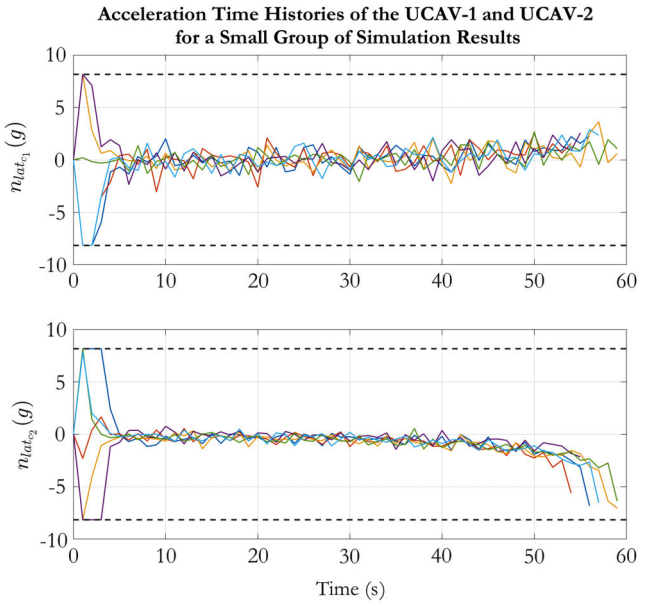


Fig. 7 Action signal time histories for a small group of Monte Carlo simulations in case 1.

RL agent generates the lateral acceleration command within the defined boundaries $n_{\text{lat}_{\text{max}}} \in [-8g, +8g]$ while meeting the operational requirements.

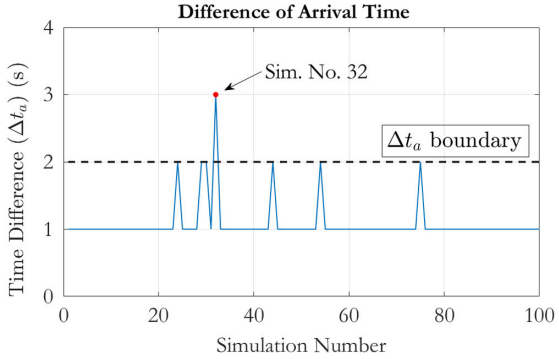


Fig. 8 Time-of-arrival difference for case 1.

Simultaneous arrival situation is checked by using the time-of-arrival difference Δt_a of the UCAVs, and the results of the Monte Carlo analysis are given in Fig. 8. According to the simulation results, the UCAV fleet meets the operational requirement for simultaneous arrival, $\Delta t_a <= 2$ s for most of the episodes. The simultaneous arrival requirement is violated only in episode 32 with $\Delta t_a = 3$ s. Despite this worst case, it is not quite far from the time-of-arrival difference requirement, and it can be concluded that the proposed system can be used effectively to saturate the enemy air defense that is placed in the target area.

There are two main reasons that increase the arrival time difference in episode 32. First, UCAV-2 starts in its outermost bound of the position interval, which is defined in the Monte Carlo analysis and it has to fly long distance to reach into the target area (dashed black line in Fig. 6). On contrary, UCAV-1 starts near its innermost position boundary and it has to fly shorter distance until the target (solid black line in Fig. 6). Hence, UCAV-1 has to increase its arrival time by performing minor maneuvers as UCAV-2 is on its way. However, because of the R_{act_i} reward given in Eq. (23), negative reward is generated while applying such kind of maneuvers to minimize the actuator usage. This restricts the maneuver ability of UCAV-1 and contributes to the increased Δt_a as the second reason.

This problem can be solved by decreasing the w_{act_i} weight in the reward function given in Eq. (17). This reduces the actuator signal usage punishment and allows auxiliary maneuvers to reduce the arrival time difference. For further investigation of this situation and its solution, the RL agent is retrained by using updated actuation

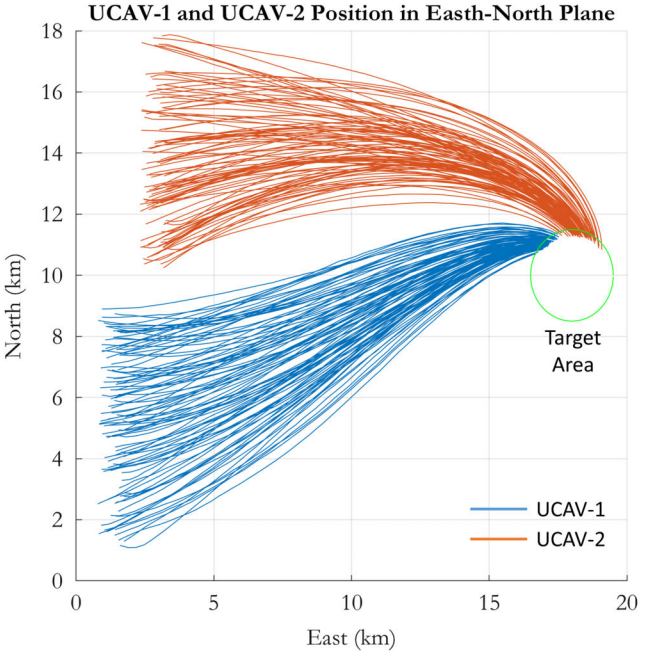


Fig. 9 UCAVs' position in east–north plane for case 1 with reduced w_{act_i} weight.

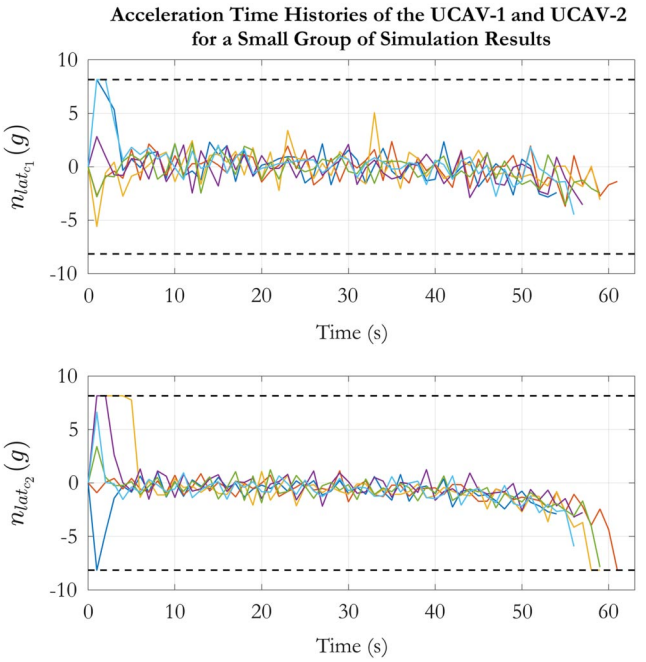


Fig. 10 Action signal time histories for a small group of Monte Carlo simulations in case 1 with reduced w_{act_i} weight.

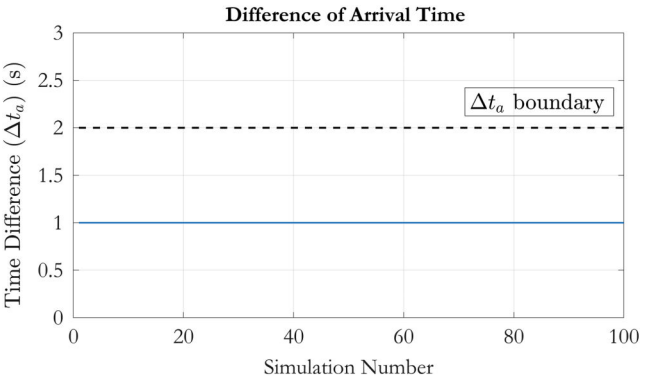


Fig. 11 Time-of-arrival difference for case 1 with reduced w_{act_i}.

signal weight as $w_{\text{act}_i} = 0.1$ and an additional 100-run Monte Carlo analysis is performed. Position of the UCAVs in the north–east plane for the whole analysis is given in Fig. 9, and actuator signal time histories for a small group of simulation results are given in Fig. 10. Difference of time-of-arrival Δt_a for case 1 with reduced actuation signal weight w_{act_i} is given in Fig. 11. Here, unlike case 1 with high w_{act_i}, it is obvious that reduction of the actuation signal weight in the reward function ensures the simultaneous arrival, and $\Delta t_a \leq 2$ s requirement is satisfied in all of the episodes of the Monte Carlo analysis.

C. Case 2: Simultaneous Arrival with NFZs

In case 2, the proposed system is trained for the warfare environment that contains enemy assets such as SAMs, artilleries, and radars. These assets are represented as NFZs. In this case, it is desired to obtain simultaneous arrival into the target area while considering collision avoidance and minimal action signal usage. As in case 1, the PPO algorithm is used in the training process and the learning curve is given in Fig. 12.

Position data of the UCAVs on east–north plane are shown in Fig. 13. Here, it is shown that, for most of the episodes in the Monte Carlo analysis, the UCAV fleet is able to reach into the target area with minimal occupancy in the NFZs. However, it is obvious that in some of the episodes of the Monte Carlo analysis, UCAVs may

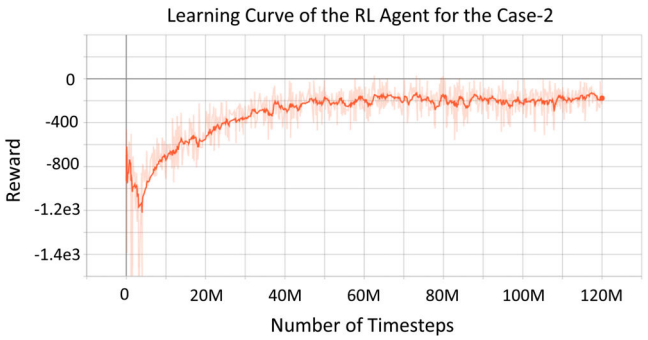


Fig. 12 Learning curve of the RL agent for case 2.

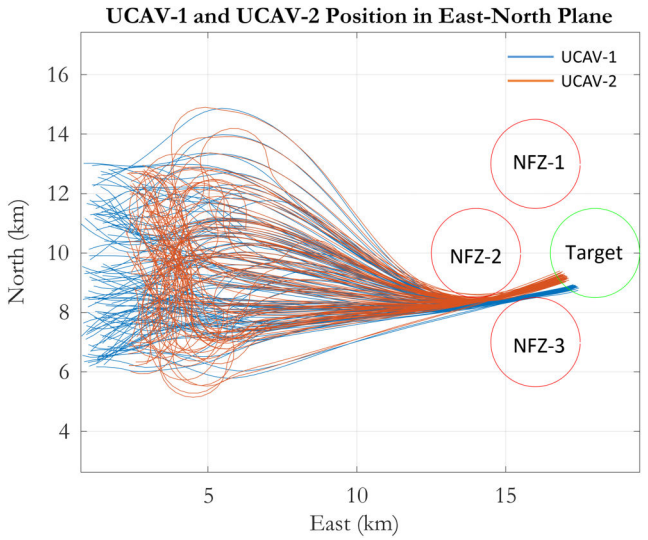


Fig. 13 UCAVs' position in east–north plane for case 2.

occupy the NFZ, which decreases the overall mission success probability. This is a problem-specific situation and can be handled by adjusting the reward function weights in Eqs. (17) and (18). For example, if the simultaneous arrival is strictly required for the operation and a narrow time interval is defined such as $\Delta t_a \leq 1$ s, the agent may generate flight paths that violate the NFZ to arrive into the target area simultaneously while considering collision avoidance and control signal usage limits. If the NFZ occupation is strictly prohibited by increasing w_{n_i} weight of the ith UCAV, the UCAV will not violate the NFZ restriction. However, it may decrease the simultaneous arrival performance. Such kind of tradeoffs should be considered carefully while defining the aerial mission requirements.

Position of the UCAVs and corresponded action signal time histories of UCAV-1 and UCAV-2 are given in Fig. 14, where R and U are used as abbreviations of run and UCAV. For example, $R1U1$ stands for the position of UCAV-1 in simulation number 1. Similarly, $R1U2$ stands for the position of UCAV-2 in simulation number 1. Here, it is shown that the RL agent generates action signal while considering the lateral acceleration bound defined in the training phase. As in case 1, position and control signal time histories are given just for five episodes of the Monte Carlo analysis to provide clarity of the figure.

In case 2, it is also important to check the possible collision situations in the Monte Carlo analysis. Hence, time history of the UCAV-to-UCAV distance is investigated in Fig. 15. The collision situations, in which the collision avoidance requirement is violated $d_{a_{1_{a_2}}} < d_{ca}$, are also shown in this figure as red lines. Here, it is observed that collision occurs in 5 of 100 episodes in the Monte Carlo analysis, which indicates 95% success from the collision avoidance point of view.

Difference of time of arrival of the UCAVs into the target area is given in Fig. 16 for each episodes. In the training process, allowable

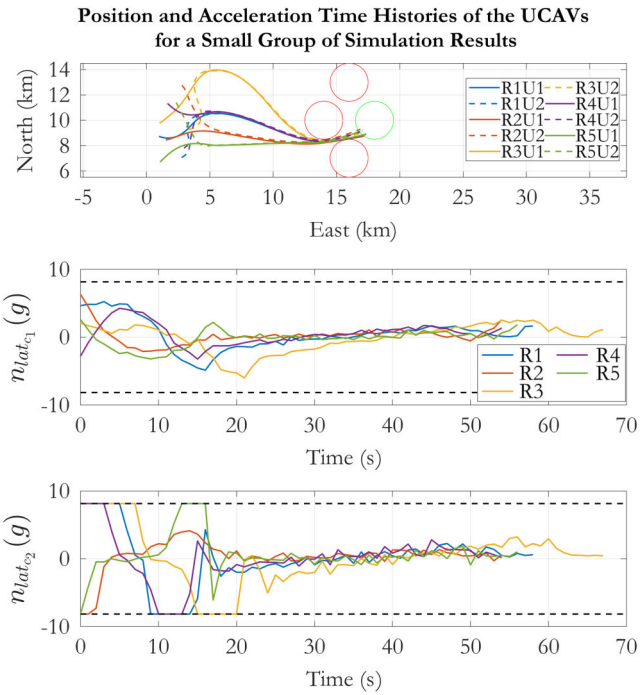


Fig. 14 Position and action signal time histories for a small group of Monte Carlo simulations in case 2.

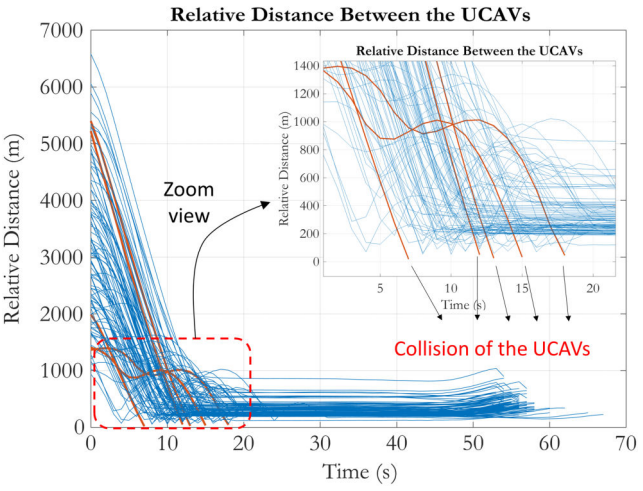


Fig. 15 Relative distance between the UCAVs for case 2.

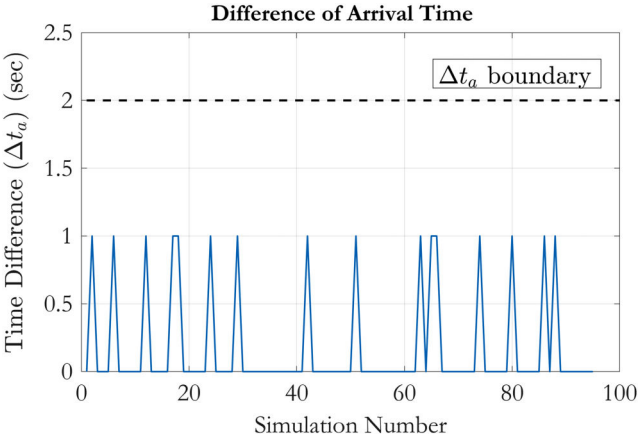


Fig. 16 Time-of-arrival difference for case 2.

maximum time-of-arrival difference is defined as $\Delta t_a <= 2$ s. Here, it is important to note that all of the collision-free flight routes provide desirable time-of-arrival difference, $\Delta t_a <= 1$ s. Combining the obtained time-of-arrival difference and number of collision-free episodes, it can be inferred that the proposed method is able to generate feasible flight paths while meeting the simultaneous arrival requirement.

D. Comparison Studies

After developing the proposed algorithm and simulating it in several case studies, it is important to compare its performance with a baseline path planning method to obtain a quantitative insight about its capabilities. For this purpose, a particle swarm optimization (PSO)-based path planning algorithm is implemented, which is a widely used population-based evolutionary algorithm originally proposed by Kennedy and Eberhant [24]. In this method, movement of a swarm particles is simulated to solve the optimization problem in a multidimensional search space. Candidate solution is represented by the position of the particles. In every iteration of the optimization process, velocity magnitudes of each particles are calculated by using the personal and social influence of the particles in Eq. (33) and particle position is updated by using Eq. (34) [25].

$$\boldsymbol{v}_{t+1} = \omega \boldsymbol{v}_t + c_1 \boldsymbol{r}_1 (\boldsymbol{b}_t - \boldsymbol{x}_t) + c_2 \boldsymbol{r}_2 (\boldsymbol{g}_t - \boldsymbol{x}_t) \tag{33}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_{t+1} \tag{34}$$

where $\boldsymbol{v}$ is velocity vector, $\boldsymbol{x}$ is position vector, $\boldsymbol{b}$ is local best position, $\boldsymbol{g}$ is global best position of the particles, and $\boldsymbol{r}_1, \boldsymbol{r}_2 \in [0, 1]$ are random numbers, and ω, c_1, c_2 are inertia weight, personal learning coefficient, and global learning coefficient, respectively. Cost function used in the PSO algorithm L^{pso} includes violation cost L_v (collision and NFZ violation) and simultaneous arrival cost L_{sa} as given in Eq. (35).

$$L^{\text{pso}} = L_v + L_{sa} \tag{35}$$

where L_v contains data about the total path length that violates the NFZ and/or collision avoidance requirements (i.e., total path length passes through the NFZ and the safe area of the UCAVs). L_{sa} contains the information about the difference of total path length of the UCAVs and indicates arrival time difference at the target area as the UCAVs fly at the equal ground speed.

For comparison studies of the proposed RL-based and PSO-based cooperative path planning algorithms, an additional 500-run Monte Carlo analysis is performed based on case 2 given in the Sec. IV by using these methods. In this analysis, ±5% NFZ position uncertainty is considered to investigate the system performance in the presence of environmental uncertainty. Analysis results are given in Table 4.

If there is no uncertainty in the position of the NFZ, the proposed method has better performance when compared with the PSO-based path planning algorithm in terms of NFZ and simultaneous arrival performance metrics, i.e., P_{nfz}, P_{sa}. However, the P_{ca} value of the proposed algorithm is slightly higher than the PSO, which means that the collision avoidance requirement $d_{a_{1a_2}} > r_{\text{safe}}$ is not met in several simulation episodes. It does not mean that a collision occurs, but it is not safe to fly closer that the $r_{\text{safe}} = 50$ m. Here, it can be stated that the centralized RL path planning algorithm produces flight paths that meet the simultaneous arrival and NFZ restrictions at the cost of increased collision buffer zone intrusion.

Table 4 Five-hundred-run Monte Carlo analysis for the PSO-based and RL-based path planning methods

		Centralized RL	
Metrics	PSO	Without uncertainty	With ±5% NFZ uncertainty
P_{nfz}	26.780	4.071	27.199
P_{ca}	0.0	0.674	0.725
P_{sa}	0.432	0.154	0.476
t_e	≈27 s	≈0.13 s	≈0.13 s

Table 5 Results of 500-run Monte Carlo analysis for the RL-based path planning method in the presence of wind effects

Metrics	No wind	±10 m/s wind	±20 m/s wind
P_{nfz}	4.071	6.924	17.184
P_{ca}	0.674	0.814	1.074
P_{sa}	0.154	0.195	0.342

In addition to the comparison of the path planning algorithms based on the defined performance metrics, it is also important to evaluate their execution time t_e to solve the problem. Although the PSO-based method has better collision avoidance capability, it takes approximately 27 s to obtain the optimal solution as it solves the whole optimization problem in each simulation episode. This is an undesirable situation for the real-time applications. On the other hand, the execution time of the proposed algorithm is approximately 0.13 s to obtain the solution, which means that the proposed algorithm is able to generate a desirable flight path 200 times faster than the PSO-based method.

When the NFZ position uncertainty is inserted into the simulation environment, it affects the system performance negatively and increases the defined metrics. However, it still has similar performance with the PSO-based method in terms of NFZ and simultaneous arrival metrics. Although a slight increment is observed in the collision avoidance metric, it is still able to generate flight paths for the simultaneous arrival mission in approximately 0.13 s.

E. External Disturbance Effects

External disturbances such as wind directly affect mission performance, and they should be considered in the path planning phase. Hence, in this part of the study, the proposed algorithm is evaluated in the presence of wind effects. The trained RL agent is used to generate flight paths in defined wind conditions while satisfying the NFZ, collision avoidance, and simultaneous arrival restrictions. The 500-run Monte Carlo analysis results are given in Table 5 for three different conditions: a) without wind, b) with ±10 m/s wind on north and east directions, and c) with ±20 m/s wind on north and east directions. Here, it is shown that even in the severe wind conditions, the proposed algorithm is able to generate flight paths with lower detection probability by the ground radar systems (i.e., lower P_{nfz}) and better simultaneous arrival capability (i.e., lower P_{sa}) than the baseline PSO-based path planning algorithm. The PSO-based algorithm still has slightly better performance in terms of collision buffer zone intrusion ability.

V. Conclusions

In this study, 2D path planning in a warfare environment for the UCAV fleet is performed by using the RL. Enemy assets such as SAMs, artilleries, and radars are represented as NFZs. Under several assumptions, the point-mass mathematical model structure is used to represent the kinematics of the UCAVs. Relative geometry between the UCAV agents and enemy assets is derived and integrated into the observation vector. The general structure of the RL-based centralized path planning algorithm is described, and observation and action vectors are explained. Operational and system requirements such as collision avoidance, simultaneous arrival, and minimal control usage are defined and embedded into the reward function. The PPO algorithm is used in the training phase of the RL agent, and hyperparameters are given for reproducibility of the study. The proposed method is applied in two different cases, and system performance is evaluated by using Monte Carlo analysis. Simulation results of case 1, which does not contain NFZ requirements, indicate that the proposed system is able to generate adequate flight routes for the UCAV fleet to obtain simultaneous arrival in the target area. After modifying the reward function weights, difference of time of arrival

meets with the defined specification $\Delta t_a <= 2$ s in almost all of the episodes of the Monte Carlo analysis. Then, the proposed method is applied in case 2, which is generated for a more difficult scenario. Unlike case 1, it contains NFZs to simulate the enemy assets such as SAMs and AAAs. Simulation results of case 2 indicate that the proposed method is able to generate collision-free flight routes in the presence of NFZs while meeting the simultaneous arrival and lateral acceleration command signal bound to obtain the saturated attack in the target area. In five episodes of the 100-run Monte Carlo simulation, high collision probability is observed between the UCAVs. The difference of time of arrival meets the defined requirement, i.e., $\Delta t_a <= 2$ s, in the 95 successful collision-free episodes of the Monte Carlo simulations. That indicates 95% success in generation of the flight routes while meeting the defined requirements.

For evaluation purposes of the proposed method, the PSO-based cooperative path planning algorithm is implemented and quantitative comparison study is performed by using the developed performance metrics for collision avoidance, NFZ avoidance, and simultaneous arrival requirements. Also, NFZ position uncertainty and wind effects are inserted into the simulation environment. Then, Monte Carlo analyses are performed to evaluate the robustness of the proposed method in the presence of environmental uncertainties and external disturbances. The analysis results indicate that the proposed cooperative path planning method outperforms the baseline PSO-based application in terms of most of the performance metrics and execution time. Although it may generate flight paths with a little bit higher NFZ violation score, this could be restored by updating the reward function weights according to aerial mission requirements.

This study contributes to the literature from two aspects. First, to the authors' best knowledge, this is the first time that a feasible and tractable RL-based centralized path planning methodology is developed for UCAV fleets. For example, in comparison to typical PSO-based methods, the RL-based approach provides the real-time ability for the fleet to replan in face of dynamic and counter-attacking/defending threats. Second, in contrast to current approaches, the proposed methodology provides the capability to simultaneously consider key operational constraints such as simultaneous arrival and collision avoidance requirements while considering NFZ restrictions and system limitations such as lateral acceleration command limit of the UCAVs. For example, typical approaches such as PSO-based methods consider a much limited subset of these restrictions and thus have applicability to only some aspects of real-life scenarios. Considering both contribution facets, the proposed approach not only provides means for real-life applicable cooperative operation capability based on fundamental terms such as closing speed and approximated time-to-go information, but also provides a real-time approximation to a highly nonlinear and large-scale UCAV fleet optimization problem.

Our current work is focused on extending the approach for the 3D path planning problem including further challenging cases such as dynamic obstacles and communication limits. As such, decentralized RL structures and new cooperative performance metrics are being explored to achieve distributed implementation capability across the UCAV fleet. In addition, to increase the scalability of the algorithm and decrease the system complexity, multi-agent RL (MARL) approach will also be implemented and cooperation ability will be investigated.

In addition, we are studying on designing of the optimized reward function that affects the performance and training duration of the RL agent. Also, sensitivity analysis has been developed for the training process to investigate the effects of the reward function component (i.e., reward and weight elements) values on the overall system and training performance. Increasing the uncertainty level on the NFZ positions and handling with the dynamic threats such as aircrafts and portable ground weapon systems will also be investigated in the future works.

Acknowledgment

This work is supported in part by the Engineering and Physical Sciences Research Council (Grant No. EP/V026763/1).

References

[1] Tsourdos, A., White, B., and Shanmugavel, M., *Cooperative Path Planning of Unmanned Aerial Vehicles*, Vol. 32, Wiley, Hoboken, NJ, 2010, Chap. 1.

[2] Eun, Y.-J., and Bang, H., "Cooperative Control of Multiple UCAVs for Suppression of Enemy Air Defense," *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, AIAA Paper 2004-6529, 2004.

[3] Gu, X., Shen, L., Chen, J., Zhang, Y., and Zhang, W., "A Virtual Motion Camouflage Approach for Cooperative Trajectory Planning of Multiple UCAVs," *Mathematical Problems in Engineering*, Vol. 2014, Jan. 2014, Paper 748974.

[4] Chen, Y. F., Everett, M., Liu, M., and How, J. P., "Socially Aware Motion Planning with Deep Reinforcement Learning," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Inst. of Electrical and Electronics Engineers, New York, 2017, pp. 1343–1350.

[5] Semnani, S. H., Liu, H., Everett, M., de Ruiter, A., and How, J. P., "Multi-Agent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, 2020, pp. 3221–3226. https://doi.org/10.1109/LRA.2020.2974695

[6] Challita, U., Saad, W., and Bettstetter, C., "Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, Vol. 18, No. 4, 2019, pp. 2125–2140. https://doi.org/10.1109/TWC.2019.2900035

[7] Julian, K. D., and Kochenderfer, M. J., "Distributed Wildfire Surveillance with Autonomous Aircraft Using Deep Reinforcement Learning," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1768–1778. https://doi.org/10.2514/1.G004106

[8] Zhou, X., Wu, P., Zhang, H., Guo, W., and Liu, Y., "Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning," *IEEE Access*, Vol. 7, Nov. 2019, pp. 165,262–165,278. https://doi.org/10.1109/ACCESS.2019.2953326

[9] Sadhu, A. K., and Konar, A., *Multi-Agent Coordination: A Reinforcement Learning Approach*, Wiley, Hoboken, NJ, 2020, Chap. 1.

[10] Foerster, J., Assael, I. A., De Freitas, N., and Whiteson, S., "Learning to Communicate with Deep Multi-Agent Reinforcement Learning," *Advances in Neural Information Processing Systems*, Vol. 29, 2016, pp. 2137–2145.

[11] Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J., "Deep Decentralized Multi-Task Multi-Agent Reinforcement Learning Under Partial Observability," arXiv preprint arXiv:1703.06182, 2017.

[12] Zhang, K., Yang, Z., and Başar, T., "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," arXiv preprint arXiv:1911.10635, 2019.

[13] Sun, C., Shen, M., and How, J. P., "Scaling Up Multiagent Reinforcement Learning for Robotic Systems: Learn an Adaptive Sparse Communication Graph," arXiv preprint arXiv:2003.01040, 2020.

[14] Omidshafiei, S., Kim, D.-K., Liu, M., Tesauro, G., Riemer, M., Amato, C., Campbell, M., and How, J. P., "Learning to Teach in Cooperative Multiagent Reinforcement Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, AAAI Press, Palo Alto, CA, 2019, pp. 6128–6136.

[15] Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T., "Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents," *Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 80, edited by J. Dy, and A. Krause, PMLR, Stockholmsmassan, Stockholm, 2018, pp. 5872–5881.

[16] Wang, R. E., Everett, M., and How, J. P., "R-maddpg for Partially Observable Environments and Limited Communication," arXiv preprint arXiv:2002.06684, 2020.

[17] Zhao, W., Fang, Z., and Yang, Z., "Four-Dimensional Trajectory Generation for UAVs Based on Multi-Agent Q Learning," *Journal of Navigation*, Vol. 73, No. 4, 2020, pp. 874–891. https://doi.org/10.1017/S0373463320000016.

[18] Yuksek, B., Demirezen, U., and Inalhan, G., "Development of UCAV Fleet Autonomy by Reinforcement Learning in a Wargame Simulation Environment," *AIAA Scitech Forum and Exposition 2021*, AIAA Paper 2021-0175, 2021.

[19] Bøhn, E., Coates, E. M., Moe, S., and Johansen, T. A., "Deep Reinforcement Learning Attitude Control of Fixed-Wing Uavs Using Proximal Policy Optimization," *2019 International Conference on Unmanned*

Aircraft Systems (ICUAS), Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 523–533.

[20] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P., "High-Dimensional Continuous Control Using Generalized Advantage Estimation," arXiv preprint arXiv:1506.02438, 2015.

[21] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," arXiv preprint arXiv: 1707.06347, 2017.

[22] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton Univ. Press, Princeton, NJ, 2012, Chap. 9.

[23] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., "Decentralized Optimization, with Application to Multiple Aircraft Coordination," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 2002, pp. 1147–1155.

[24] Kennedy, J., and Eberhart, R., "Particle Swarm Optimization," *Proceedings of ICNN'95—International Conference on Neural Networks*, Vol. 4, Inst. of Electrical and Electronics Engineers, New York, 1995, pp. 1942–1948.
https://doi.org/10.1109/ICNN.1995.488968

[25] Roberge, V., Tarbouchi, M., and Labonté, G., "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning," *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 1, 2012, pp. 132–141.
https://doi.org/10.1109/TII.2012.2198665

H. Oh
Associate Editor



https://dspace.org/

Cooperative planning for an unmanned combat aerial vehicle fleet using reinforcement learning

Yuksek, Burak; Demirezen, Mustafa Umut; Inalhan, Gokhan; Tsourdos, Antonios

2021-07-07
American Society of Mechanical Engineers (ASME)
https://doi.org/10.2514/1.I010961
Yuksek B, Demirezen MU, Inalhan G, Tsourdos A. (2021) Cooperative planning for an unmanned combat aerial vehicle fleet using reinforcement learning, Journal of Aerospace Information Systems, Volume 18, Issue 10, October 2021, pp. 739-750.

Downloaded from a DSpace repository. https://dspace.org/