https://doi.org/10.31891/csit-2025-2-20
UDC 629.735.33:004.896

VELYCHKO Maksym, KYSIL Tetiana
Khmelnytskyi National University

REINFORCEMENT LEARNING METHOD FOR AUTONOMOUS FLIGHT PATH PLANNING OF MULTIPLE UAVS

This study aims to develop a reinforcement learning method for autonomous flight path planning of multiple UAVs under real-world conditions with limited observations and multiple conflicting optimization objectives. The research proposes a multi-agent reinforcement learning approach based on Proximal Policy Optimization (PPO) combined with centralized training and decentralized execution (CTDE). Additionally, a recurrent neural network (RNN) layer is integrated into the critic and actor networks to address partial observability. The reward function is designed to balance time efficiency, safety, and area coverage. Experimental results demonstrate that the proposed method significantly outperforms independent learning approaches in terms of reward accumulation, convergence speed, and decision stability. The CTDE architecture with RNN-enhanced critics proved effective in handling the challenges of multi-agent coordination and partial observability. The trained model enables real-time trajectory planning in three-dimensional environments, surpassing traditional optimization methods. The novelty lies in the application of a multi-agent PPO architecture enhanced by RNNs under CTDE for solving real-time multi-objective optimization problems in UAV path planning. A customized reward structure was developed to simultaneously optimize safety, time, and coverage objectives without retraining. The developed method enables efficient and reliable online trajectory planning for UAV groups, making it applicable in surveillance, search and rescue, and exploration missions where rapid and adaptive decision-making is essential.

Keywords: multiple UAVs, path planning, reinforcement learning, centralized training, decentralized execution, multi-agent systems, PPO algorithm, RNN, CTDE architecture.

ВЕЛИЧКО Максим, КИСІЛЬ Тетяна
Хмельницький національний університет

МЕТОД НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ АВТОНОМНОГО ПЛАНУВАННЯ ТРАЄКТОРІЇ ПОЛЬОТУ ГРУПИ БПЛА

Метою роботи є розробка методу навчання з підкріпленням для автономного планування траєкторій польоту групи БПЛА в умовах обмеженої видимості середовища та конфліктних цілей оптимізації. Запропоновано багатоагентний підхід навчання з підкріпленням на основі алгоритму проксимальної оптимізації політики (PPO) з використанням архітектури централізованого навчання та децентралізованого виконання (CTDE). Для покращення роботи в умовах часткових спостережень інтегровано рекурентну нейронну мережу в структури акторів і критиків. Розроблено спеціалізовану функцію винагороди, яка враховує показники безпеки, швидкості досягнення цілей та площі покриття території. Результати експериментів показали перевагу запропонованого методу над незалежним навчанням за критеріями швидкості збіжності, стабільності стратегії та величини отриманої винагороди. Структура CTDE із рекурентними мережами дозволила ефективно вирішити проблеми координації між БПЛА та неповної інформації про середовище. Модель забезпечує автономне планування траєкторій у реальному часі у тривимірних середовищах. Наукова новизна полягає в поєднанні методів глибокого навчання з підкріпленням, рекурентних нейронних мереж та архітектури CTDE для вирішення задач багатокритеріальної оптимізації в умовах часткової доступності даних. Розроблений підхід дозволяє підвищити ефективність групової навігації БПЛА, зокрема в сферах розвідки, пошуково-рятувальних операцій і моніторингу, де важливими є автономність, швидкість реагування і надійність.

Ключові слова: планування траєкторій, навчання з підкріпленням, централізоване навчання, децентралізоване виконання, багатоагентні системи, алгоритм PPO, RNN, архітектура CTDE.

Introduction

Nowadays, autonomous flight path planning for multiple UAVs has become a critical challenge due to dynamic environments, limited sensing capabilities, and the need for real-time decision-making. Traditional algorithms struggle with scalability and adaptability in such settings. Reinforcement learning (RL), particularly deep reinforcement learning (DRL), has demonstrated remarkable success in decision-making tasks under uncertainty. However, multi-agent systems introduce additional complexity, including decentralized information and conflicting objectives. This study proposes a reinforcement learning approach based on Proximal Policy Optimization (PPO) and a Centralized Training with Decentralized Execution (CTDE) paradigm. The method addresses the challenges of partial observability by integrating recurrent neural networks into the actor-critic architecture and formulates a reward function that balances time, safety, and coverage criteria. The goal is to enable autonomous, cooperative, and efficient flight path planning for a group of UAVs operating in real-world conditions.

Related works

Most previous studies on multi-UAV path planning focused on intelligent optimization methods, particularly evolutionary algorithms like Particle Swarm Optimization (PSO) [1–3]. Shao et al. [4] improved PSO for better convergence and obstacle avoidance, while Evan et al. [5] adapted PSO for unknown environments, and Ajeil et al. [6] proposed a hybrid PSO method optimizing path smoothness. However, swarm intelligence algorithms struggle with scalability and real-time performance, making them less suitable for reconnaissance missions.

The rise of Deep Reinforcement Learning (DRL) opened new possibilities for decision-making in dynamic, partially observable environments [7–9]. Reinforcement learning excels where traditional algorithms fail, offering better generalization and real-time inference speed.

Challenges in multi-UAV settings include limited perception, dynamic state changes, and coordination. Fully distributed learning architectures are inefficient for solving Multi-Agent Path Finding (MAPF) problems. Centralized training with decentralized execution (CTDE), first proposed by Lowe [10], addresses these issues by using additional information during training. Jose et al. [11] applied CTDE in vehicle routing, achieving near-optimal solutions, while Marc et al. [12] and Wang et al. [13] demonstrated improvements in UAV conflict resolution and dynamic routing using similar frameworks.

Reward design is critical in reinforcement learning for multi-objective tasks. Simple reward structures risk "reward hacking" [14], while overly complex rewards hurt generalization. The common practice is to transform multi-objective problems into single-objective optimization via weighted reward functions, achieving near-optimality. Li [15] and Xu [16] proposed advanced DRL frameworks to solve multi-objective problems in robotic control and continuous optimization, relevant for UAV path planning where time, safety, and coverage constraints must be considered.

Purpose

Route planning for multiple UAVs can be viewed as a Multi-Agent Path Finding (MAPF) problem, which is a model used to find the optimal path for multiple agents from start positions to destinations without conflicts. In fact, MAPF is a relatively complex optimisation problem with a common goal. The state space of this problem grows exponentially with the number of agents, and it has been proven to be NP-hard. In reconnaissance missions, UAV groups must not only avoid dangerous areas and safely reach target points, but also cover a larger area in a shorter time. However, time and coverage area are in conflict with each other because these are multi-objective optimisation problems, and we must find a trade-off between two or more conflicting objectives to make an optimal decision. It is not possible to find a solution that will achieve optimal fulfilment of all objectives, so for a multi-objective optimisation problem, a set of non-compromised solutions is usually used, which is called a "Pareto solution set".

Proposed technique

Due to the development of deep reinforcement learning (DRL), researchers are actively studying its application for trajectory planning and navigation of UAV groups. Unlike traditional algorithms, DRL performs better in unknown and dynamic environments, providing fast inference and good generalisation in real-time tasks.

This study takes into account partial observations of a group of UAVs, which makes it difficult to make optimal decisions due to the lack of global information. Since the actions of individual UAVs change the state of the environment, incomplete data reduces the effectiveness of learning. To coordinate the actions, a centralised learning architecture with decentralised execution is proposed, which has proven to be effective in multi-agent tasks.

Particular attention is paid to the reward function: too simple a function impairs learning, and too complex reduces generalisation. A multi-factor reward approach with appropriate weighting is optimal. When planning routes for UAVs, time, safety, and coverage are taken into account.

An improved multiagent algorithm based on PPO, a model-free reinforcement learning method that provides adaptability and generalisation, is proposed. The centralised PPO critic network coordinates UAVs through joint observations, and the actor network determines actions. The addition of a recurrent neural network allows historical information to be taken into account to compensate for partial observations. A joint reward function is also developed to learn the optimal policy. After training, each UAV acts based on local data.

During reconnaissance missions, multiple UAVs must plan routes to target points in real time, avoiding collisions and taking into account time and coverage. Autonomous trajectory planning in such conditions is a distributed decision-making problem with partial observations and multi-objective optimisation. Since all UAVs work cooperatively to achieve a common goal, the problem is modelled as a Decentralised Partially Observable Markov Decision Process (Dec-POMDP). Dec-POMDP allows multiple agents to make decisions based on local observations without knowledge of the global state, while being rewarded for a shared long-term benefit. Since the optimal solution of this distributed model has double exponential complexity, reinforcement learning is used to approximate the solution.

In reinforcement learning, the agent optimises its policy by interacting with the environment and receiving rewards. There are methods based on values and policies: the former are difficult to apply in continuous action spaces, and the latter suffer from low efficiency. To solve this problem, actor-critical algorithms have been developed, where an actor generates actions and a critic evaluates their quality, which increases the efficiency of learning. Popular examples: DDPG, PPO, and A3C.

In cooperative multi-agent environments (Dec-POMDP), simple independent learning of agents is inefficient due to the instability of the environment. The MADDPG algorithm solves this problem through centralised critic training, where the input is based on the joint observations of all agents. Actors act on the basis of local information. This architecture of centralised training and decentralised execution (CTDE) ensures the stability and efficiency of multi-agent learning, as in COMA.

In this study, the proximal policy optimisation (PPO) algorithm is chosen for UAV control, which combines the efficiency of the policy gradient with the stability of optimisation due to the constraint of policy changes. PPO is based on an actor-critical architecture and scales well for problems with a continuous action space, making it suitable for real-time UAV trajectory planning.

Typically, policy gradient algorithms have high variability due to excessive policy updates, which leads to learning instability. PPO solves this problem through a special objective function that limits the updates, ensuring that the policy is gradually adapted over several iterations. The algorithm takes into account the difference between the old and new network when updating the parameters. The basis of PPO is the expected reward gradient, which is defined as:

$$\nabla\bar{R}(\tau)=\mathrm{E}_{\tau\sim\pi\theta(\tau)}\left[A^{\pi}\left(s_t,a_t\right)\nabla\log p_{\theta}\left(a_t\mid s_t\right)\right], \tag{1}$$

where π_{θ} – is the policy parameterised by the vector θ;
$A^{\pi}\left(s_t,a_t\right)$ – the function of superiority.

The preference function determines how much a particular action is better than the average in a given state and is calculated as:

$$A\left(s,a\right)=Q\left(s,a\right)-V\left(s\right), \tag{2}$$

where $Q^{\pi}\left(s,a\right)$ – is the expected total reward after performing the action$a$ a in the state s;
$V^{\pi}\left(s\right)$ – expected remuneration from the state of s according to the policy of π.

To ensure the stability of updates, PPO uses the importance sampling technique:

$$\mathrm{E}_{x\sim p}\left[f(x)\right]=\mathrm{E}_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right], \tag{3}$$

which allows us to reassess expectations under the new policy using the data collected from the old one.

The main feature of PPO is the use of clipping to stabilise learning. This prevents excessive changes in the policy. The corresponding constrained objective function is as follows:

$$\mathrm{L}^{CLIP}(\theta)=\mathrm{E}_t\left[\min\left(\left(\frac{\pi_{\theta}(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)}\right)\hat{A}_t,\mathrm{clip}\left(\frac{\pi_{\theta}(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)},1-\text{т},1+\text{т}\right)\hat{A}_t\right)\right], \tag{4}$$

where $\hat{A}_t$ – is the estimate of the preference function at step t;
т – defines the acceptable limit of policy changes.

This approach strikes a balance between exploring new actions and maintaining an effective strategy, which is especially important for controlling many UAVs in a real-world environment.

Based on the PPO algorithm, the CTDE (Centralised Training with Decentralised Execution) architecture training method was applied and a multi-agent PPO algorithm was developed in a multi-agent environment. Compared to a single-agent environment, the input to the critic network is the joint observation of several UAVs, which is equivalent to the operation of a centralised controller. This approach allows each drone to receive more relevant information for decision-making. The actor network is updated by maximising the following objective function:

$$\mathrm{L}(\theta)=\frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\left[\min\left(r_{\theta,i}^{k}A_i^{k},\mathrm{clip}\left(r_{\theta,i}^{k},1-\text{т},1+\text{т}\right)A_i^{k}\right)+\sigma\cdot S_{\pi}\right], \tag{5}$$

where $r_{\theta,i}^{k} = \dfrac{\pi_{\theta}(a_i^k \mid o_i^k)}{\pi_{\theta_{\text{old}}}(a_i^k \mid o_i^k)}$ – is the probability ratio between the new and old policies;

A_i^k – the function of superiority;

т – is a restriction parameter;

S_{π} – the entropy of politics, which facilitates the exploration of new actions.

The critic's network, in turn, is updated by minimising the loss of the value function, which is defined as follows:

$$\mathrm{L}(\phi) = \frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\max\left[\left(\mathrm{V}_{\phi}(s_i^k) - \mathrm{R}_i\right)^2, \left(\mathrm{clip}\left(\mathrm{V}_{\phi}(s_i^k), \mathrm{V}_{\phi_{\text{old}}}(s_i^k) - \varepsilon, \mathrm{V}_{\phi_{\text{old}}}(s_i^k) + \varepsilon\right) - \mathrm{R}_i\right)^2\right], \tag{6}$$

where R_i^k – is a target value (for example, a win or a return estimate);

$\mathrm{V}_{\phi}(s)$ – the state value estimated by the current critic network.

The weights of the two neural networks (actor and critical) are updated in each episode. The update process is shown in figure 1.

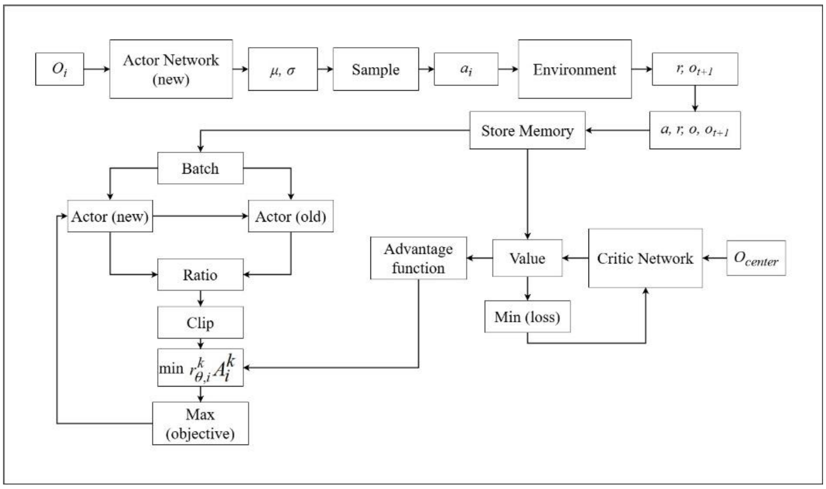

Fig.1. Scheme of updating the weights of the actor-critic network in each episode

The actor network, receiving the state of the environment o_t as input, generates normal distribution parameters μ, σ, from which the action a_t is selected. After the action is performed, the environment returns the reward r_t and the new state o_{t+1}. The collected data is saved to the memory buffer, after which a data batch is formed and the advantage function is calculated. At the same time, the critical network is updated to minimise losses. Next, the ratio between the old and new policies is calculated, the policy change limitation operation is applied (clipping), and the maximum of the objective function is selected. This helps to stabilise learning and avoid large policy updates.

One of the main problems of autonomous route planning for a group of UAVs is partial observation, due to which agents have limited information. To compensate for this, recurrent neural networks (RNNs) that store historical data are used. The first successful combination of RNNs and reinforcement learning was implemented in the Deep Recurrent Q-Network (DRQN) (figure 2), where one of the linear layers of the DQN was replaced by an RNN, which allowed it to work without stacking frames.

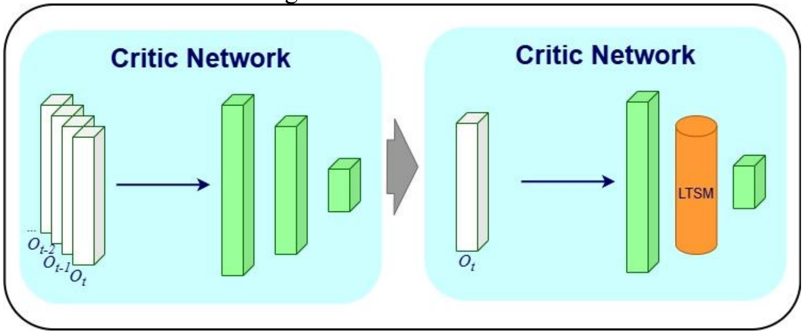

Fig.2. Adding a recurrent layer (LSTM) to the Critical Network architecture

In this study, an RNN layer is added to the PPO algorithm to process sequences of historical observations. Since classical RNNs have a short-term memory limitation due to gradient decay, the Long Short-Term Memory (LSTM) architecture is used to solve this problem (figure 3). It uses a gate mechanism that allows:

- forgetting unnecessary information (forget gate);
- remember new information (input gate);
- output relevant information to the next step (output gate).

Thanks to this structure, LSTM effectively retains important time dependencies and prevents gradients from damping.

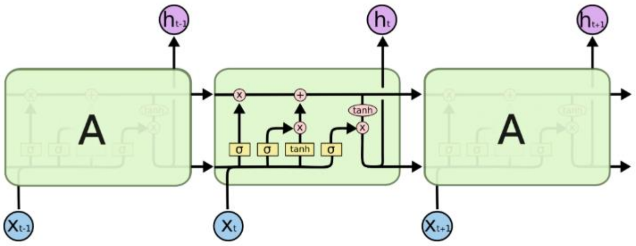

Fig.3. The structure of LSTM: at time t, $X_{(t)}$ is the input signal, C_t is the cell state, h_t is the hidden state

At each time step, the input x_t is processed together with the hidden state $h_{(t-1)}$ and the cellular state $C_{(t-1)}$, which allows generating updated values of h_t and $C_t.\sigma$

The τ packets are used to update the parameters of the actor and critic networks to maximise $\mathrm{L}(\theta)$ and minimise $\mathrm{L}(\varphi)$ using gradient descent:

$$\tau = [s_t, o_t, a_t, r_t, s_{t+1}, o_{t+1}, a_{t+1}, \ldots] \tag{7}$$

We add a layer to the LSTM network, and two elements $h_{(t,\pi)}$ and $h_{t,V}$ 1 are added to τ, which are changed to:

$$\tau = [s_t, o_t, h_{t,\pi}, h_{t,V}, a_t, r_t, s_{t+1}, o_{t+1}, h_{t+1,\pi}, h_{t+1,V}, a_{t+1}, \ldots], \tag{8}$$

where $h_{t,\pi}$ та $h_{t,V}$ – are the hidden states of the actor network and the critic network, respectively, at time t.

The goal is to maximise the loss function for the actor network:

$$\mathrm{L}(\theta) = \frac{1}{Bn}\sum_{i=1}^{B}\sum_{k=1}^{n}\left[\min\left(r_{\theta,i}^{k} A_i^k, \mathrm{clip}(r_{\theta,i}^{k}, 1-\text{т}, 1+\text{т}) A_i^k\right) + \sigma \cdot S_\pi\right], \tag{9}$$

where $r_{\theta,i}^{k}$ – is the probability ratio between the new and old policies;

A_i^k – the function of superiority;

т – the cutoff value;

S_π – policy that stimulates research.

The study analyses the functional purpose of the network of critics and actors in multi-agent reinforcement learning based on the architecture of centralised training with decentralised execution (CTDE). It is established that the critic network acts as a central controller, processing complete observations, so adding an artificial neural network layer to its structure can significantly improve the efficiency of the model in conditions of partial observation.

The actor network implements the policy of an individual agent, and the addition of an RNN layer has a smaller impact on its performance. Experimental results confirmed the feasibility of using the CTDE architecture to solve this problem.

Autonomous route planning for a group of UAVs is considered as a multi-objective optimisation problem that takes into account time, coverage area, and system safety. Optimisation involves the selection of decision variables in a discrete space.

This approach is similar in nature to the "action selection" task in reinforcement learning. The combination of "offline learning" and "online decision-making" within the framework of deep reinforcement learning allows for real-time implementation of multi-objective optimisation. Thus, deep reinforcement learning methods are an effective tool for solving multi-objective optimisation problems in systems with several UAVs, providing high generalisability of the built models.

The reward function was developed based on the principles of multi-objective optimisation and previous knowledge in the field of navigation by decomposing the overall task into a number of sub-tasks. The joint reward function was formed taking into account the constraints on safety, execution time and territory coverage. The reward function has the following form:

$$r_{\text{total}} = \alpha \cdot r_{\text{timecost}} + \beta \cdot r_{\text{security}} + \gamma \cdot r_{\text{coverage}}, \tag{10}$$

where $r_{\text{security}} = \sum \left|\text{distance}(UAV_i - UAV_j)\right| - \left|\text{distance}(UAV_i - \text{target}_j)\right|$ – is a component that helps to avoid collisions between UAVs by controlling the distances between them, and encourages them to approach target points;

$r_{\text{coverage}} = \sum \text{new area}_{UAV_i}$ – reward for exploring new areas of the territory, which encourages UAVs not only to achieve goals, but also to actively explore the environment;

r_{timecost} – a component that helps to minimise the time to reach target points by reducing the number of steps.

The coefficients α, β, γ are set taking into account the priority of the relevant criteria and can be changed depending on the mission requirements. Thanks to this structure of the reward function, the system can adapt to different scenarios while maintaining an optimal balance between efficiency, safety and coverage. This approach makes it possible to turn a multi-criteria problem into a single-criteria problem with the possibility of further adjusting the aggressiveness of the agent's strategy without complete retraining. A multi-objective optimisation problem can be solved by means of multi-objective reinforcement learning.

To solve the problem of multicriteria policy optimisation, a gradient multicriteria approach is used. This approach is aimed at maximising the weighted sum of rewards, where the weight vector ω determines the importance of each of the objective functions. This leads to a change in the appearance of the policy gradient.

The functional of the objective of a multi-criteria policy can be represented as follows:

$$\mathrm{J}\ (\theta,\omega) = \omega^{\bullet}\ F(\pi) = \sum_{i=1}^{m} \omega_i f_i(\pi) = \sum_{i=1}^{m} \omega_i \mathrm{J}\ _i^{\pi}, \tag{11}$$

where θ – is the policy settings;

$\omega \in \square^{m}$ – a vector of weights for each target;

$\mathrm{J}\ _i^{\pi}$ – goal functionality for the *i* task.

The gradient of this function by policy parameters is as follows:

$$\nabla_{\theta}\mathrm{J}\ (\theta,\omega) = \sum_{i=1}^{m} \omega_i \nabla_{\theta}\mathrm{J}\ _i(\theta) \tag{12}$$

Which can then be represented as a mathematical expectation:

$$\begin{aligned} &= \mathrm{E}\left[\sum_{t=0}^{T} \omega^{\bullet}\ A^{\pi}(s_t, a_t)\nabla_{\theta}\pi_{\theta}(a_t \mid s_t)\right] \\ &= \mathrm{E}\left[\sum_{t=0}^{T} A_{\omega}^{\pi}(s_t, a_t)\nabla_{\theta}\pi_{\theta}(a_t \mid s_t)\right], \end{aligned} \tag{13}$$

where A^{π} – is an advantage function;

A_{ω}^{π} – weighted preference, which takes into account all goals with appropriate weights.

Experiments

In this study, the PyTorch library was used to build a three-level actor neural network and critique the PPO algorithm. Centralised training with decentralised execution (CTDE) was implemented to coordinate the UAV's actions. The key difference between centralised and independent training is the composition of the input data for the critic network.

In the experiment, the local observations of all UAVs were aggregated into a single high-dimensional vector, which formed a joint observation for the critic's network (O_{centre}). The actor network received individual observations O_i of each UAV (see figure 4).

Compared to independent training, four control experiments were conducted. The network architecture included three layers: the first and third layers were fully connected, and the second layer was a recurrent LSTM layer. To verify the feasibility of LSTM, a variant of the architecture without the recurrent layer was created, which allowed for a comparative analysis and confirmation of the advantages of the CTDE architecture.

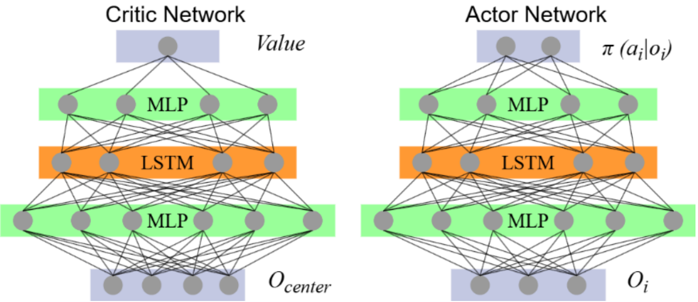

Fig.4. The network of actors and the network of critics, as well as all the agents that share these networks

The proximal policy optimisation produces random policies, which means that the outputs of the actor network are μ, σ, which are the mathematical expectation and variance of a Gaussian distribution, and the output action is a random sample from this Gaussian distribution.

In the process of training agents using the PPO method, we used the fixed hyperparameters shown in table 1.

Table 1

Neural network parameters and PPO agent training

№	Parameter	Meaning
1	Episode	625
2	Episode length	200
3	Rollout thread	16
4	Clip	0,2
5	Discount	0,99
6	Entropy coefficient	0,1
7	Buffer size	500
8	Batch size	32
9	FC layer dim	128
10	RNN hidden dim	64
11	Activation	ReLU
12	Optimizer	Adam

After performing one million training steps and analysing the experimental results, it was found that the PPO algorithm implemented on the basis of centralised training with decentralised execution demonstrated higher efficiency compared to independent training in the task of autonomous trajectory planning for a group of UAVs.

As can be seen from the results shown in figure 5, fully independent and distributed learning methods have difficulty achieving adequate performance in multi-UAV tasks. The introduction of the CTDE architecture significantly improved the efficiency: the value of the reward function became positive, and the performance increased even as the number of UAVs increased.

This confirms the feasibility of using CTDE to solve distributed tasks. The central controller coordinates the actions of all UAVs, but with more than six UAVs, the efficiency decreases due to the increase in the size of the joint observation space.

Experiments were conducted to test whether the addition of a recurrent neural network (RNN) layer can solve the problem of training many UAVs in the face of incomplete information.

The experimental results (figure 6) showed that adding an RNN layer to the actor and critic networks significantly improves the model's performance. Including the RNN layer in the critics' network provided an improvement, but the convergence process was slower. Adding an RNN layer to the actor network alone did not lead to a significant performance gain and did not solve the problem of partial observations when using multiple UAVs.

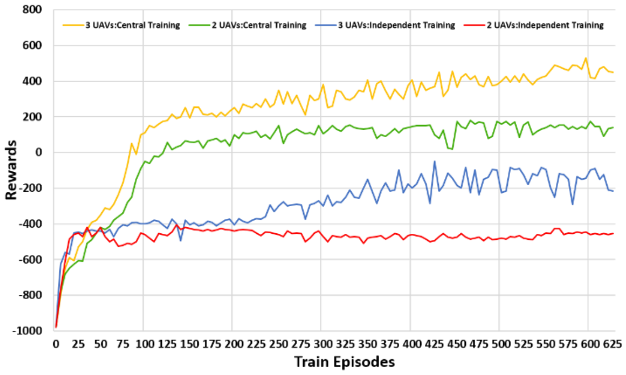

Fig.5. Comparison of CTDE and independent architecture

These results confirm that in the CTDE architecture, the critic network functions as a central controller, coordinating the UAV's actions, and adding an RNN layer to this network helps to compensate for the problem of incomplete information.

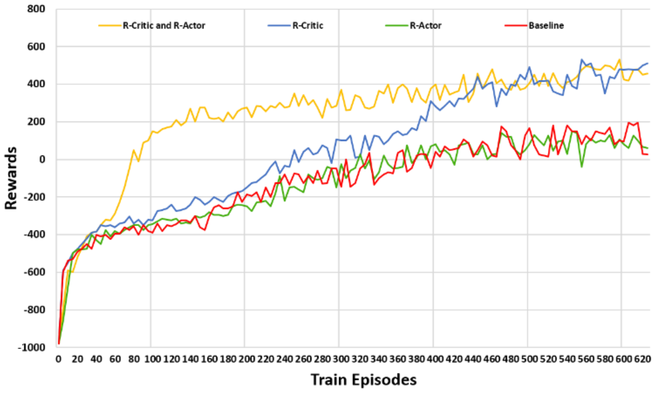

Fig.6. Model performance after adding an RNN layer to the network of actors and critics

Reinforcement learning demonstrates significantly faster execution speeds than traditional swarm intelligence algorithms, making it suitable for solving real-time problems (figure 8). After training, the parameters of the neural network were fixed, ensuring the stability of the model's behaviour during execution. Solving the navigation task with this model showed higher and more stable rewards compared to other approaches.

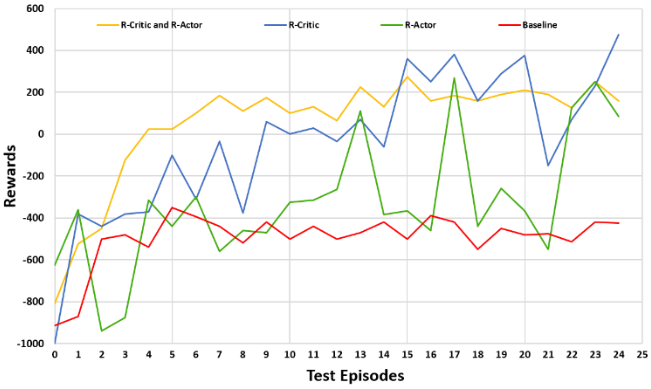

Fig.8. Comparison of the average reward of different architectures at the testing stage

Conclusions

The study successfully demonstrates that reinforcement learning methods, specifically PPO with CTDE architecture and recurrent neural enhancements, significantly improve the efficiency and reliability of multi-UAV autonomous flight path planning. The centralized critic provides comprehensive coordination based on collective observations, while decentralized actors ensure flexible execution. The developed reward structure effectively balances multiple mission objectives, leading to better overall system performance. Experimental results confirm that the proposed approach outperforms traditional and independent learning methods, highlighting its potential for deployment in complex, dynamic environments requiring real-time decision-making and collaboration among multiple agents.

References

1. Tang J., Duan H., Lao S. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: a comprehensive review. *Artificial Intelligence Review*. 2022.

2. Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms / Y. Wang et al. *IEEE Access*. 2019. Vol. 7. P. 105086–105099.

3. An Accurate UAV 3-D Path Planning Method for Disaster Emergency Response Based on an Improved Multiobjective Swarm Intelligence Algorithm / Y. Wan et al. *IEEE Transactions on Cybernetics*. 2022. P. 1–14.

4. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization / S. Shao et al. *ISA Transactions*. 2020. Vol. 97. P. 415–430.

5. Collision-Free Autonomous Robot Navigation in Unknown Environments Utilizing PSO for Path Planning / E. Krell et al. *Journal of Artificial Intelligence and Soft Computing Research*. 2019. Vol. 9, no. 4. P. 267–282.

6. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm / F. H. Ajeil et al. *Applied Soft Computing*. 2020. Vol. 89. P. 106076.

7. Deep-Sarsa Based Multi-UAV Path Planning and Obstacle Avoidance in a Dynamic Environment / W. Luo et al. *Lecture Notes in Computer Science*. Cham, 2018. P. 102–111.

8. Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning / C. H. Liu et al. *IEEE Transactions on Mobile Computing*. 2020. Vol. 19, no. 6. P. 1274–1285.

9. Multi-UAV Path Planning for Wireless Data Harvesting With Deep Reinforcement Learning / H. Bayerlein et al. *IEEE Open Journal of the Communications Society*. 2021. Vol. 2. P. 1171–1187.

10. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the Advances in Neural Information Processing Systems. Long Beach. CA. USA. 4–9 December 2017. Volume 30.

11. Vera J. M., Abad A. G. Deep Reinforcement Learning for Routing a Heterogeneous Fleet of Vehicles. *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Guayaquil, Ecuador, 11–15 November 2019. 2019.

12. Brittain M., Wei P. Autonomous Separation Assurance in An High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach. *2019 IEEE Intelligent Transportation Systems Conference - ITSC*, Auckland, New Zealand, 27–30 October 2019. 2019.

13. Cooperative Reinforcement Learning Aided Dynamic Routing in UAV Swarm Networks / Z. Wang et al. *ICC 2022 - IEEE International Conference on Communications*, Seoul, Korea, Republic of, 16–20 May 2022. 2022.

14. Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; Amodei, D. Reward learning from human preferences and demonstrations in atari. In Proceedings of the Advances in Neural Information Processing Systems. Montreal, Canada. 3–8 December 2018. Volume 31.

15. Li K., Zhang T., Wang R. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Transactions on Cybernetics*. 2020. P. 1–12.

16. Xu, J.; Tian, Y.; Ma, P.; Rus, D.; Sueda, S.; Matusik, W. Prediction-guided multi-objective reinforcement learning for continuous robot control. In Proceedings of the International Conference on Machine Learning. Virtual. 13–18 July 2020. P. 10607–10616.

Maksym Velychko **Максим Величко**	Master Student of Computer Engineering & Information Systems Department, Khmelnytskyi National University e-mail: vmaks230@gmail.com	Магістрант кафедри комп’ютерної інженерії та інформаційних систем, Хмельницький національний університет
Tetiana Kysil **Тетяна Кисіль**	Candidate of Physical and Mathematical Sciences, Associate Professor of Computer Engineering & Information Systems Department, Khmelnytskyi National University https://orcid.org/0000-0002-4094-3500 e-mail: kysil_tanya@ukr.net	Кандидат фізико-математичних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет