drones

MDPI

Article

A Dual Aircraft Maneuver Formation Controller for MAV/UAV Based on the Hybrid Intelligent Agent

Luodi Zhao [1,2,3], Yemo Liu [4], Qiangqiang Peng [4] and Long Zhao [1,2,3,*]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; luodizhao@buaa.edu.cn
[2] Digital Navigation Center, Beihang University, Beijing 100191, China
[3] Science and Technology on Aircraft Control Laboratory, Beihang University, Beijing 100191, China
[4] Beijing Aerospace Automatic Control Institute, Beijing 100854, China
[*] Correspondence: buaa_dnc@buaa.edu.cn

Abstract: This paper proposes a hybrid intelligent agent controller (HIAC) for manned aerial vehicles (MAV)/unmanned aerial vehicles (UAV) formation under the leader–follower control strategy. Based on the high-fidelity three-degrees-of-freedom (DOF) dynamic model of UAV, this method decoupled multiple-input-multiple-output (MIMO) systems into multiple single-input-single-output (SISO) systems. Then, it innovatively combined the deep deterministic policy gradient (DDPG) and the double deep Q network (DDQN) to construct a hybrid reinforcement learning-agent model, which was used to generate onboard desired state commands. Finally, we adopted the dynamic inversion control law and the first-order lag filter to improve the actual flight-control process. Under the working conditions of a continuous S-shaped large overload maneuver for the MAV, the simulations verified that the UAV can achieve accurate tracking for the complex trajectory of the MAV. Compared with the traditional linear quadratic regulator (LQR) and DDPG, the HIAC has better control efficiency and precision.

Keywords: MAV/UAV; formation control; hybrid reinforcement learning; hybrid intelligent agent

check for updates

Citation: Zhao, L.; Liu, Y.; Peng, Q.; Zhao, L. A Dual Aircraft Maneuver Formation Controller for MAV/UAV Based on the Hybrid Intelligent Agent. *Drones* **2023**, *7*, 282. https://doi.org/10.3390/drones7050282

Academic Editor: Shiva Raj Pokhrel

Received: 28 February 2023
Revised: 12 April 2023
Accepted: 18 April 2023
Published: 22 April 2023

1. Introduction

Aiming at increasingly fast-paced and high-intensity air combat, the use of MAVs as combat operations leaders with a certain number of UAVs as wingers to form a hybrid formation of UAV/MAV has become the development trend for future air confrontations. Among them, the two-aircraft formation consisting of an MAV and a UAV is one of the most typical combat styles. In MAV/UAV formations, the unmanned system must be able to share information and carry out cooperative operations with the manned systems across systematic boundaries [1]. The Fast Lightweight Autonomy (FLA) Program by the Defense Advanced Research Projects Agency (DARPA) has developed an advanced algorithm that enables an MAV or a UAV to operate autonomously without a human operator, the Global Positioning System (GPS), or any data resources. DARPA's Lifelong Learning Machines (L2M) Project also aims to develop new machine learning methods that enable unmanned systems to continuously adapt to new environments and remember what they have learned [2]. Meanwhile, the U.S. Air Force's Loyal Wingman Program aims to enhance the autonomy of UAVs and improve their combat capabilities in complex war environments [3]. Moreover, the recently proposed Skyborg program is working on the combination of manned and unmanned combat aerial vehicles. Therefore, improving the capability of autonomous flight control has become an important direction for the development of future UAV technology.

One of the research hotspots of UAV autonomous control capability is the formation flight-control problem [4]. In terms of the traditional design of the formation controller, Ref. [5] proposed a sliding mode controller for MAV/UAV formation flights based on a layered architecture. However, it makes extensive simplifications on the strong nonlinear dynamic model of MAV/UAVs, and was only validated by simulations for flat trajectories. Ref. [6] considered sensor noise and developed a leader–follower formation PID controller for multi-robots, which can achieve better performance in limiting position deviations. Furthermore, Ref. [7] proposed a parallel approach control law for fixed-wing UAV formations under the leader–follower strategy. Ref. [8] referred to an idea of multi-channel decoupling that split the MIMO system into multiple SISO systems and used sliding mode control to track the reference trajectory, which can be further applied to formation-control problems. Refs. [9,10] proposed a consensus-based multiple aircraft cooperative formation control method, but the consensus theory analysis was highly dependent on the linearized dynamic model, which limited its further application in a complex nonlinear dynamic system. Refs. [11,12] developed a formation controller where the commands were generated independently of the dynamic model, decreasing the control precision in extreme working conditions. Refs. [13,14] considered the confrontation situation and adopted predefined maneuver strategy collections, taking typical maneuvers as the basic units and building a collection of maneuver strategies with free combinations of various basic units. However, due to the model uncertainty and non-cooperative environment, this method hardly dealt with complex working conditions. Therefore, the intelligent agent method has become a novel research trend because of its weak model dependence and strong ability in terms of strategy exploration. Refs. [15,16] adopted deep neural networks to learn aircraft-maneuvering strategies and made progress in enhancing the autonomous maneuvering capability of UAVs. However, UAV formation control is a high-dimensional dynamical control problem with tightly coupled variables. When traditional neural networks learn such complex behaviors, they cause problems such as low training efficiency and difficulty in stable convergence [17]. Among the novel neural networks, the double deep Q network (DDQN) algorithm has shown good performance in control problems with discrete action sets by fitting the value functions of state actions through neural networks [18–20], but it cannot be applied to control problems with continuous variables. Based on the deterministic policy gradient (DPG) algorithm, DeepMind proposed the deep deterministic policy gradient (DDPG) algorithm which is proven to perform well on many kinds of continuous control problems [21–23]. However, in the field of aircraft control, the large variation in the angle of attack commands will increase the load on the attitude control loop [24]. Meanwhile, when it comes to complex tasks with multiple continuous control variables problems, DDPG has problems with unstable networks and low exploration efficiency [25–28]. For the above dilemma, some scholars have turned to hybrid reinforcement learning methods in recent years. By adding discrete "meta-actions" to continuous control problems, Ref. [29] partially solved the reinforcement learning traps and improved exploration efficiency. The experiments verified its superiority to the traditional continuous strategy algorithm in some cases. [30] proposed the parametrized deep Q-network for the hybrid action space without approximation or relaxation, which provides a reference for solving the hybrid control problem.

Based on the above analysis, it is obvious that the formation controller must be able to better adapt to complex flight conditions in future confrontation situations, e.g., continuous large overload maneuvers for the MAV, etc. Therefore, inspired by [29], we propose a hybrid reinforcement intelligent agent controller based on the decoupling of multi-channels, which can effectively solve the problem of formation-tracking under continuous maneuvering conditions. It should be emphasized that when designing controllers based on artificial intelligent methods, especially when the reinforcement learning controller is directly applied to the generation of underlying flight-control commands, the lack of flight dynamic constraints can easily bring about problems. Due to the lack of dynamic constraints, the attitude control system cannot quickly track the commands, leading to flight instability. Therefore, this paper introduced the dynamic inversion controller and the first-order lag filter to the hybrid reinforcement learning agent to enhance the smoothness and executability of control commands.

In summary, the main contributions of this paper are as follows:

(1) A hybrid intelligent agent was designed based on the novel concept of "meta-action" to further enhance formation control performance. The hybrid intelligent agent combined DDPG and DDQN according to the specific formation control targets;

(2) The framework of the HIAC was developed that combined the dynamic inversion controller and the first-order lag filter with the hybrid intelligent agent to effectively overcome the common drawbacks of reinforcement learning;

(3) The superiority of the HIAC method was validated with experiments of nominal conditions. Monte Carlo simulations with different initial conditions were then conducted to verify the adaptability of the HIAC.

The organization of this paper is as follows: Section 2 establishes the UAV dynamic model and formation-control targets. Section 3 designs the novel formation controller HIAC based on the DDPG/DDQN hybrid intelligent agent. The dynamic inversion controller and first-order lag filter are introduced to the framework of the HIAC as well. Section 4 conducts the experiments of nominal conditions and 100 Monte Carlo simulations with varying initial conditions. Finally, we summarize the research conclusion of this paper in Section 5.

2. Mathematical Modeling

2.1. UAV Dynamic Model

The main concern in dual aircraft formation flights is the real-time position, velocity, and attitude of the two aircraft, so it is necessary to establish a dynamic model of the UAV according to the forces on the mass as shown in Figure 1. To simplify the problem, the constraints flight envelope is ignored.

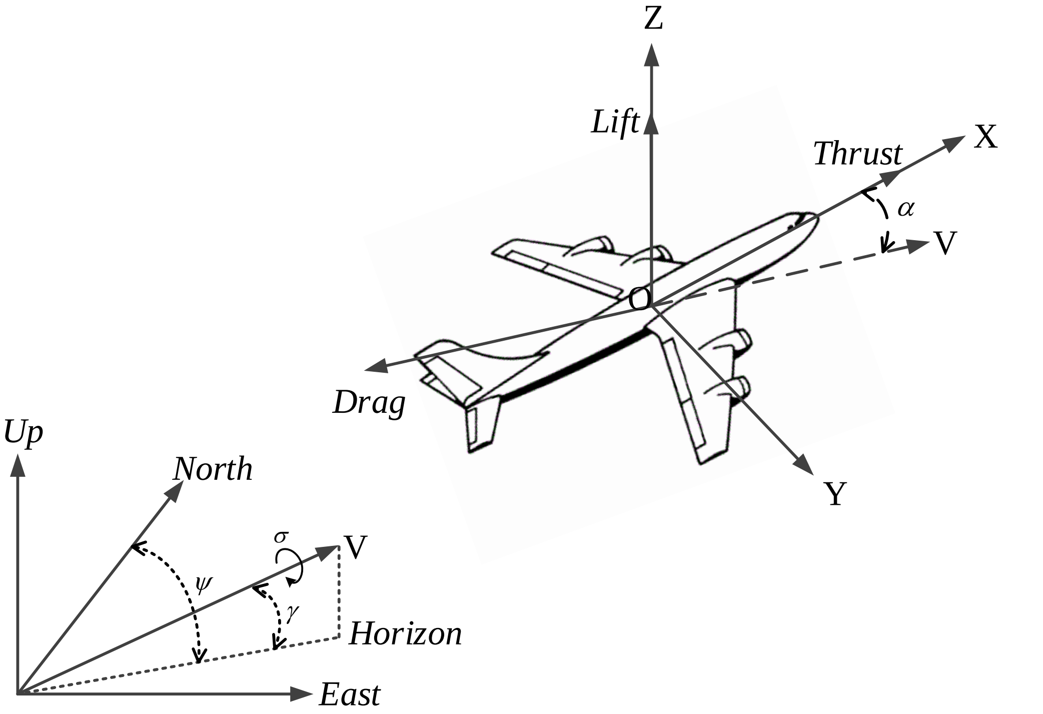


Figure 1. The forces on the center of gravity of the aircraft.

In the ground inertial coordinate system $o-xyz$, V is the UAV flight velocity. γ and ψ are the flight path angle and flight azimuth angle, respectively. The flight adopts the Bank-To-Turn (BTT), which is considered to have no sideslip. α is the attack angle, and σ is the bank angle. The engine thrust and drag of the aircraft are denoted by T and D, respectively. n is the normal overload of the UAV in the velocity coordinate system

$o - x_V y_V z_V$. Ignoring the wind disturbance in the flight, the three-degrees-of-freedom of the dynamic model for the UAV is established as follows [31–33]:

$$H = \begin{cases} \dot{x} = V\cos\gamma\sin\psi \\ \dot{y} = V\cos\gamma\cos\psi \\ \dot{z} = V\sin\gamma \\ \dot{V} = (T-D)/m - g\sin\gamma \\ \dot{\gamma} = g(n\cos\sigma - \cos\gamma)/V \\ \dot{\psi} = -gn\sin\sigma/(V\cos\gamma) \end{cases}, \tag{1}$$

where m is the weight of the aircraft, which is considered constant in this paper, and g is the local gravity.

The engine thrust T can be denoted by

$$T = \eta T_{\max}, \tag{2}$$

where η is the throttle manipulator, and its range is defined as $[0,\ 1]$. $T_{\max}$ is the maximum thrust that the engine can achieve.

The air drag D consists of the parasite drag and the induced drag, which can be expressed as follows [31]:

$$D = C_{D_P}\rho V^2 S/2 + 2C_{D_I} n^2 m^2 g^2 / \left(\rho V^2 S\right), \tag{3}$$

where S is the reference area of the UAV. C_{D_P} is the parasite drag coefficient. C_{D_I} is the induced drag coefficient. ρ is the atmospheric density, which varies with the altitude of the aircraft in the stratosphere. It is calculated by [34]

$$\rho = \rho_0 \cdot e^{-z/z_0}, \tag{4}$$

where $\rho_0 = 1.225\ \mathrm{kg/m^3}$ and $z_0 = 6700$ m.

2.2. Formation Control Targets

In this paper, the formation control target of the UAV was determined based on the leader–follower formation strategy. Taking a typical dual aircraft formation flight as an example, the formation configuration of the MAV/UAV was designed as shown in Figure 2. Since the reference trajectory of the MAV as the leader aircraft is known, the flight velocity, attitude, and position can be obtained from the sensors mounted within the MAV. The winger aircraft can receive real-time flight data from the MAV through the onboard data chain and complete the trajectory tracking and formation control autonomously. During the flight, it is required that the UAV and MAV keep a specific formation throughout the whole flight, as shown in Figure 2.

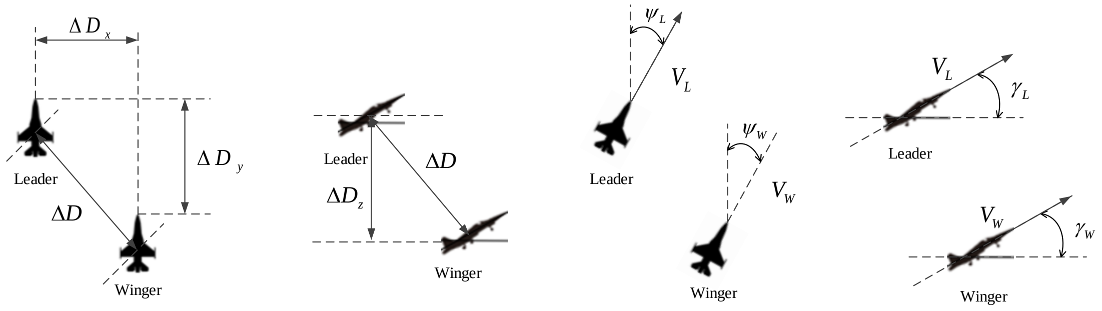


Figure 2. Dual aircraft formation for MAV/UAV.

2.2.1. Flight Velocity Control Targets

The MAV and UAV keep the same formation flight velocity. The reference velocity of the MAV is V_L, and the UAV velocity is V_W, then the velocity deviation ΔV is

$$\Delta V = |V_L - V_W|. \tag{5}$$

The MAV and UAV keep the same flight path angle in formation flight. The MAV flight path angle is γ_L, and the UAV flight path angle is γ_W, then the flight path angle deviation $\Delta\gamma$ is

$$\Delta\gamma = |\gamma_L - \gamma_W|. \tag{6}$$

The MAV and UAV keep the same flight azimuth angle in formation flight. The flight azimuth angle of the MAV is ψ_L, and the flight azimuth angle of the UAV is ψ_W, then the deviation of the flight azimuth angle $\Delta\psi$ is

$$\Delta\psi = |\psi_L - \psi_W|. \tag{7}$$

The flight velocity and attitudes of the UAV should be consistent with the MAV within an allowable error

$$\Delta V \le V_{\Delta\max},\ \Delta\gamma \le \gamma_{\Delta\max},\ \Delta\psi \le \psi_{\Delta\max}, \tag{8}$$

where $V_{\Delta\max}$, $\gamma_{\Delta\max}$, $\psi_{\Delta\max}$ represent the error thresholds of the velocity, flight path angle, and flight azimuth angle of the UAV, respectively.

2.2.2. Flight Distance Control Targets

The UAV is located around the MAV and maintains the specified formation distance. ΔD denote the distance between the MAV and the UAV in the ground inertial coordinate system. ΔD_x, ΔD_y and ΔD_z denote the spatial distance of ΔD as follows:

$$\Delta D = \sqrt{\Delta D_x{}^2 + \Delta D_y{}^2 + \Delta D_z{}^2}. \tag{9}$$

Summarily, the UAV should keep a distance larger than the safe flight distance from the MAV, which is as follows:

$$D_{\Delta\min} \le \Delta D \le D_{\Delta\max}, \tag{10}$$

where $D_{\Delta\min}$ and $D_{\Delta\max}$ represent the thresholds of the safe distance.

3. Design of the HIAC

The HIAC first adopted a DDPG/DDQN hybrid reinforcement learning method to train the agent model to generate the tracking commands. Then, we further designed a dynamic inversion controller and a first-order lag filter to construct an improved formation flight controller. Overall, the HIAC consists of three parts, i.e., desired state command solver, dynamic inversion controller, and first-order lag filter. The framework of the HIAC is shown in Figure 3.

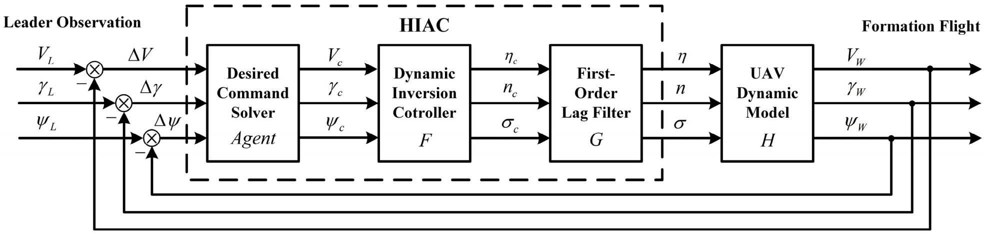


Figure 3. The framework of the HIAC.

In order to track the MAV, the HIAC adopted the current deviation between the states of the UAV and the states of the MAV, i.e., ΔV, $\Delta\gamma$ and $\Delta\psi$ as inputs, and outputs the control commands of the thrust, normal overload, and bank angle, i.e., η, n and σ. The main difference of the HIAC from other traditional controllers is that to further enhance the control accuracy, the HIAC adopted a hybrid intelligent agent as the desired command solver to generate the desired commands, V_c, γ_c and ψ_c. Then, these commands were sent to the dynamic inversion controller to generate the control commands, η_c, n_c and σ_c. Finally, the first-order lag filter further smoothed η_c, n_c and σ_c to improve the executability of these commands. The three parts will be introduced in detail as the order of the information flow.

3.1. Desired Command Solver

Learning from the idea of "meta-action", we partially discretized the control variables in the continuous control problems and developed a continuous–discrete mixed action space according to the characteristics of these control variables. Based on this process, we constructed a hybrid intelligent agent based on DDPG and DDQN to control V, γ, ψ and D of the UAV.

3.1.1. Framework of Hybrid Intelligent Agent Based on DDPG/DDQN

Based on the traditional Q-Learning algorithm, DDQN uses the neural network to fit the value function. It adopts discrete action sets to define the strategy and evaluates the Q value of the generated strategy through the Critic network. Compared with the traditional DQN algorithm [18–20], DDQN decouples the action selection strategy of the Q value and the calculation of the Q value and solves the problem of overestimation of the Q value compared with the traditional methods.

DDPG adopts the Actor–Critic network based on DQN and uses continuous action sets to define the control strategy. The model consists of the Actor–Critic network, where the Critic evaluates the actions generated by the Actor, and the Actor feeds back the evaluation results to the Critic for policy optimization [23]. More proofs and conclusions of the DDQN and DDPG can be found in [18,23], respectively.

However, the DDQN and DDPG suffer from different drawbacks when applied in practical engineering. Although the DDQN is easier to converge when compared with DDPG, it can only deal with discrete and low-dimensional action spaces. However, most of the practical targets, especially physical control targets, have continuous and high-dimensional action spaces. Moreover, even though the continuous space can be transferred into the discrete space, DDQN will generate high high-dimensional action space in this process and finally cause quite low computational efficiency. Meanwhile, although DDPG can solve the problem of continuous and high-dimensional action spaces, it is more likely to diverge than DDQN. Therefore, learning from "meta-action", we proposed a hybrid intelligent agent combining the DDQN and DDPG according to their complementary characteristics. Considering the value range and the control precision of V, γ and ψ, we adopted the idea of multi-channel decoupling to perform partial discretization of the action space. For the velocity control agent V_c, the DDPG was used to generate the set of continuous state commands. Because the value range of V_c is larger than γ_c and ψ_c, discretizing the continuous action space with high precision will lead to dimension explosion. Meanwhile, for the angle control agents γ_c and ψ_c, the DDQN is used to generate the set of discretized state commands. Combining the DDQN and DDPG can improve the capability of convergence when these two agents are trained together.

The framework of the desired commands solver was designed as shown in Figure 4. It includes three agents which process the variation of the state commands V_c, γ_c and ψ_c, respectively. Based on the decoupling principles between different agents, each agent calculates the action A_V, A_γ and A_ψ, and updates the desired state commands respectively.

The outputs are executed by the flight-control system of the UAV and fed back the rewards of each agent. The total reward function $R_{\sum}$ is expressed by

$$R_{\sum} = R_{\sum}^{(D,V)} + R_{\sum}^{(\gamma)} + R_{\sum}^{(\psi)}, \tag{11}$$

where $R_{\sum}^{(D,V)}$, $R_{\sum}^{(\gamma)}$ and $R_{\sum}^{(\psi)}$ are components of $R_{\sum}$ in each agent.

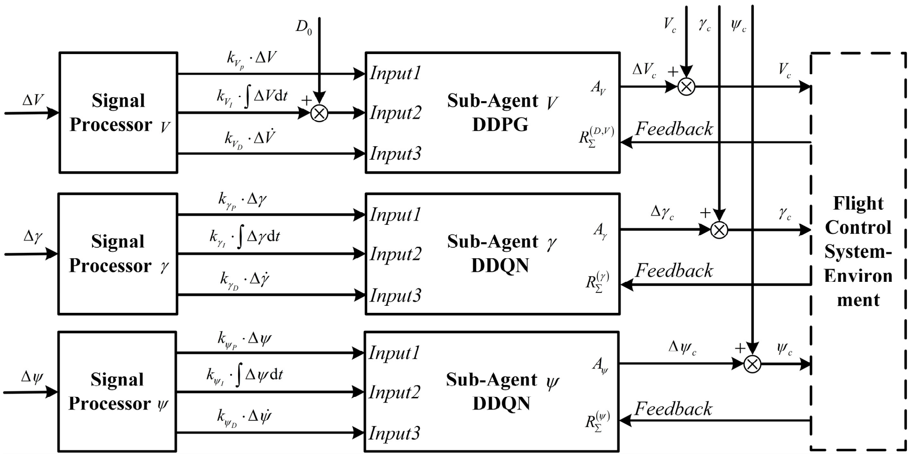


Figure 4. The framework of the desired commands solver.

To construct an intelligent agent based on the DDPG/DDQN hybrid reinforcement learning network, it was necessary to transform the trajectory tracking problem into a Markov decision process, which mainly includes three parts, i.e., the state space, the action space, and the reward function.

3.1.2. State Space S

According to the targets of formation flight control, the state space S is designed as follows:

$$S = [\Delta V, \Delta D, \Delta\dot{V}, \Delta\gamma, \int \Delta\gamma \mathrm{d}t, \Delta\dot{\gamma}, \Delta\psi, \int \Delta\psi \mathrm{d}t\ , \Delta\dot{\psi}], \tag{12}$$

where ΔD equals

$$\Delta D = \Delta D_0 + \int \Delta V \mathrm{d}t, \tag{13}$$

where ΔD_0 is the flight distance deviation between the MAV and the UAV at the initial epoch. The integral items ΔV, $\Delta\gamma$ and $\Delta\psi$ are the cumulative deviation from the initial epoch till the current epoch. $\Delta\dot{V}$, $\Delta\dot{\gamma}$ and $\Delta\dot{\psi}$ is the deviation rate of the velocity, flight azimuth angle, and flight path angle.

3.1.3. Action Space A

The action space A is defined as follows:

$$A = \left[A_V, A_\gamma, A_\psi\right], \tag{14}$$

where the action A_V denotes the correction value of the UAV velocity commands ΔV_c, the action A_γ denotes the correction value of the UAV flight path angle commands $\Delta\gamma_c$, and the action A_ψ denotes the correction value of the UAV flight azimuth angle commands $\Delta\psi_c$, i.e.,

$$\left\{ \begin{array}{c} \Delta V_c = A_V, \Delta\gamma_c = A_\gamma, \Delta\psi_c = A_\psi \\ |\Delta V_c| \leq \lambda_{V_c\max c}, |\Delta\gamma_c| \leq \lambda_{\gamma_c\max}, |\Delta\psi_c| \leq \lambda_{\psi_c\max}, \end{array} \right. \tag{15}$$

where $\lambda_{V_c\max}$, $\lambda_{\gamma_c\max}$ and $\lambda_{\psi_c\max}$ are the maximum of corrections, respectively. A_V is used to generate the set of continuous velocity commands. A_γ, A_ψ is used to generate the set of discretized angle commands. Specifically, the discretization can be further expressed as follows:

$$\begin{aligned} |\Omega_{\gamma_c}| &= 2\lceil \lambda_{\gamma_c\max}/\partial\gamma_c \rceil + 1 \\ |\Omega_{\psi_c}| &= 2\lceil \lambda_{\psi_c\max}/\partial\psi_c \rceil + 1 \end{aligned} \tag{16}$$

$$\begin{aligned} \Omega_{\gamma_c} &= \{ A_\gamma | 0, \pm\partial\gamma_c, \pm 2\partial\gamma_c, \cdots, \pm(|\Omega_{\gamma_c}| - 1)\partial\gamma_c/2, \pm\lambda_{\gamma_c\max} \} \\ \Omega_{\psi_c} &= \{ A_\psi | 0, \pm\partial\psi_c, \pm 2\partial\psi_c, \cdots, \pm(|\Omega_{\psi_c}| - 1)\partial\psi_c/2, \pm\lambda_{\psi_c\max} \}. \end{aligned} \tag{17}$$

Then, the update of desired state commands is

$$\begin{cases} V_c \leftarrow V_c + \Delta V_c \\ \gamma_c \leftarrow \gamma_c + \Delta\gamma_c \\ \psi_c \leftarrow \psi_c + \Delta\psi_c. \end{cases} \tag{18}$$

3.1.4. Reward Function $R_\sum$

According to the formation control targets of UAVs, the reward function $R_\sum$ was designed as follows:

$$R_\sum = R_P + R_N + R_C \tag{19}$$

where R_P is the reward sub-function, which gives a positive response when the flight state of the UAV meets the control targets. R_N is the penalty sub-function, which gives a negative response when the flight states exceed the allowable error of the control target. R_C is the command limiting function, which can limit the values of the control commands η_c, n_c, σ_c. More specifically, R_C can smooth the variation of the control commands to finally reduce energy consumption.

R_P is calculated by

$$R_P = 10 \times \left(\varepsilon_D^2 + \varepsilon_V^2 + \varepsilon_\gamma^2 + \varepsilon_\psi^2 \right) \tag{20}$$

where ε_D, ε_V, ε_γ, and ε_ψ are reward coefficients, which are defined as follows

$$\begin{aligned} \varepsilon_D &= \begin{cases} 1, D_{\Delta\min} \le \Delta D \le D_{\Delta\max} \\ 0, \Delta D < D_{\Delta\min} \text{ or } \Delta D > D_{\Delta\max} \end{cases}, \\ \varepsilon_V &= \begin{cases} 1 - \Delta V / V_{\Delta\max}, \Delta V \le V_{\Delta\max} \\ 0, \Delta V > V_{\Delta\max} \end{cases}, \\ \varepsilon_\gamma &= \begin{cases} 1 - \Delta\gamma / \gamma_{\Delta\max}, \Delta\gamma < \gamma_{\Delta\max} \\ 0, \Delta\gamma \ge \gamma_{\Delta\max} \end{cases}, \\ \varepsilon_\psi &= \begin{cases} 1 - \Delta\psi / \psi_{\Delta\max}, \Delta\psi < \psi_{\Delta\max} \\ 0, \Delta\psi \ge \psi_{\Delta\max} \end{cases}. \end{aligned} \tag{21}$$

R_N is calculated by

$$R_N = -100 \times \left(e_D^2 + e_V^2 + e_\gamma^2 + e_\psi^2 \right) \tag{22}$$

where e_D, e_V, e_γ, and e_ψ are penalty coefficients, which are defined as follows:

$$\begin{aligned}
e_D &= \begin{cases} 1, \Delta D < D_{\Delta\min} \text{ or } \Delta D > D_{\Delta\max} \\ 0, D_{\Delta\min} \le \Delta D \le D_{\Delta\max} \end{cases}, \\
e_V &= \begin{cases} 1, \Delta V > 2V_{\Delta\max} \\ \Delta V / V_{\Delta\max} - 1, V_{\Delta\max} \le \Delta V \le 2V_{\Delta\max} \\ 0, \Delta V < V_{\Delta\max} \end{cases}, \\
e_\gamma &= \begin{cases} 1, \Delta\gamma > 2\gamma_{\Delta\max} \\ \Delta\gamma / \gamma_{\Delta\max} - 1, \gamma_{\Delta\max} \le \Delta\gamma \le 2\gamma_{\Delta\max} \\ 0, \Delta\gamma < \gamma_{\Delta\max} \end{cases}, \\
e_\psi &= \begin{cases} 1, \Delta\psi > 2\psi_{\Delta\max} \\ \Delta\psi / \psi_{\Delta\max} - 1, \psi_{\Delta\max} \le \Delta\psi \le 2\psi_{\Delta\max} \\ 0, \Delta\psi < \psi_{\Delta\max} \end{cases}.
\end{aligned} \tag{23}$$

R_C is calculated by

$$R_C = -0.2(|\eta_c|/\eta_{\max} + |n_c|/n_{\max} + |\sigma_c|/\sigma_{\max}). \tag{24}$$

3.2. Dynamic Inversion Controller

To realize tracking of the commands of a given flight trajectory, the dynamic inversion control law was designed as follows [35]

$$\begin{aligned}
\dot{V}_c &= \varpi_V(V_c - V) \\
\dot{\gamma}_c &= \varpi_\gamma(\gamma_c - \gamma) \\
\dot{\psi}_c &= \varpi_\psi(\psi_c - \psi)
\end{aligned} \tag{25}$$

where ϖ_V, ϖ_γ, and ϖ_ψ denote the bandwidth of the controller, respectively. V_c, γ_c, ψ_c denote the desired state commands of the flight velocity, the flight path angle, and the flight azimuth angle, respectively.

Since the UAV commands follow the dynamic constraints by Equation (1), considering Equations (1), (2), and (25) yields

$$\begin{aligned}
T_c = \eta_c T_{\max} &= [D + m\varpi_V(V_c - V) + mg\sin\gamma], \\
N_\gamma &= \varpi_\gamma V(\gamma_c - \gamma)/g + \cos\gamma, \\
N_\psi &= \varpi_\psi V(\psi_c - \psi)\cos\gamma/g,
\end{aligned} \tag{26}$$

where N_γ and N_ψ denote the normal overload and lateral overload, respectively. The throttle δ_c, normal overload n_c and bank angle σ_c are selected as the control commands. Then, the UAV control command was designed as follows:

$$F = \begin{cases} \eta_c = [D + m\varpi_V(V_c - V) + mg\sin\gamma]/T_{\max} \\ n_c = \sqrt{N_\gamma{}^2 + N_\psi{}^2} \\ \sigma_c = \arctan(N_\psi/N_\gamma) \end{cases}. \tag{27}$$

Moreover, the control command must satisfy the constraints:

$$\eta_{\min} \le \eta_c \le \eta_{\max}, 0 \le n_c \le n_{\max}, |\sigma_c| \le \sigma_{\max} \tag{28}$$

where $\eta_{\min}$ and $\eta_{\max}$ is the minimum and maximum values of the throttle commands, respectively. $n_{\max}$ is the maximum value of the normal overload, and $\sigma_{\max}$ is the maximum value of the bank angle.

3.3. First-Order Lag Filter

Considering the fact that the UAV cannot instantly complete the change of the engine thrust, normal overload, and bank angle, a first-order lag filter model was constructed to simulate the delayed variation processes of these three variables:

$$G = \begin{cases} \dot{\eta} = (\eta_c - \eta)/\tau_\delta \\ \dot{n} = (n_c - n)/\tau_n \\ \dot{\sigma} = (\sigma_c - \sigma)/\tau_\sigma \end{cases}, \tag{29}$$

where η_c, n_c, σ_c represent the control commands of the throttle, normal overload, and bank angle, respectively. τ_δ, τ_n, and τ_σ represent the response time of the UAV control system accordingly.

Summarily, considering Equations (1), (27), and (29), the UAV flight process can be presented by the control equations as follow:

$$\begin{cases} F(V_c, \gamma_c, \psi_c)^{\mathrm{T}} = [\eta_c, n_c, \sigma_c]^{\mathrm{T}} \\ G(\eta_c, n_c, \sigma_c)^{\mathrm{T}} = [\dot{\eta}, \dot{n}, \dot{\sigma}]^{\mathrm{T}} \\ H(\eta, n, \sigma)^{\mathrm{T}} = [V, \gamma, \psi]^{\mathrm{T}} \end{cases}. \tag{30}$$

Equation (30) reveals the calculation process from the desired control commands to the actual control commands. It is clear that the premise to realize the formation flight is to acquire the desired control commands of the UAV V_c, γ_c, ψ_c under the specific formation strategy. Then, the ultimate flight trajectory can be obtained by the Runge–Kutta method.

4. Simulation Validation

4.1. Simulation Design

Based on the 3-DOF dynamic model in this paper, the MAV was designed to make a complex maneuver and provide the reference trajectory and control commands, accordingly. Under the leader–follower formation strategy, the UAV adopts the HIAC, DDPG, and LQR to track the MAV and keep the dual aircraft formation, respectively. LQR is a commonly used guidance method for tracking multi-state trajectories in aerospace engineering and it has been validated by extensive flight tests [36,37]. Therefore, we compared the proposed method with LQR and DDPG to verify its superiority in the following Sections 4.2 and 4.3. The design of DDPG is described in Section 3.1.

First, the experiment of nominal conditions was conducted to analyze the superiority of the proposed method in detail. Meanwhile, the initial values greatly affect the performance of the reinforcement learning models. Therefore, the generalization ability of the model was required to be fully verified. Then, 100 Monte Carlo experiments were conducted to verify the adaptability of this method to different initial conditions.

The simulations were conducted by Matlab2021a and the 3-DOF dynamic model was built by Simulink. The total simulation time was T, the simulation interval was ΔT, and the specific experimental parameters are shown in Table 1.

The training methods of DDPG and DDQN refer to [18,23], respectively. Learning rate, max episode, discount factor, and experience buffer length were set as the same for both DDPG and DDQN. In addition, the batch size of DDPG was set to 256, and the batch size of DDQN was set to 64. The specific parameters are shown in Table 2.

Table 1. The experimental parameter settings.

Parameters	Settings	Parameters	Settings
T (s)	50	$\eta_{\min}$	0
ΔT (s)	0.1	$\eta_{\max}$	1
$T_{\max}$ (lb)	25,600	$n_{\max}$	6
m (kg)	14,470	$\sigma_{\max}$ (rad)	$\pi/2$
g (m/s^2)	9.81	$D_{\Delta\max}$ (m)	600
S (ft^2)	400	$D_{\Delta\min}$ (m)	100
C_{D_P}	0.02	$V_{\Delta\max}$ (m/s)	50
C_{D_I}	0.1	$\psi_{\Delta\max}$ (rad)	0.2
τ_δ (s)	0.6	$\gamma_{\Delta\max}$ (rad)	0.2
τ_n (s)	0.5	$\lambda_{V_c\max}$ (m/s)	50
τ_σ (s)	0.5	$\lambda_{\gamma_c\max}$ (rad)	$\pi/2$
ϖ_V (s)	0.3	$\lambda_{\psi_c\max}$ (rad)	$\pi/2$
ϖ_γ (s)	0.2	$\partial\gamma_c$ (rad)	$\pi/180$
ϖ_ψ (s)	0.2	$\partial\psi_c$ (rad)	$\pi/180$

Table 2. The training parameters of DDPG/DDQN.

Parameters	Settings
Learning Rate	0.0001
Max Episode	25,000
Batch Size (DDPG)	256
Batch Size (DDQN)	64
Discount Factor	0.99
Experience Buffer Length	1×10^6

4.2. Basic Principles of LQR

The implementation of LQR mainly includes three parts: linearization of the motion model, design of the tracking controller for the reference trajectory, and solution of the feedback gain matrix.

By linearizing the dynamic model of the UAV in Equation (1) with small deviations, the linear system can be obtained as follows:

$$\dot{X} = AX + Bu. \tag{31}$$

Equation (31) can be expressed by

$$\begin{bmatrix} \delta\dot{x} \\ r\delta\dot{y} \\ \delta\dot{z} \\ \delta\dot{V} \\ \delta\dot{\gamma} \\ \delta\dot{\psi} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} \\ A_{61} & A_{62} & A_{63} & A_{64} & A_{65} & A_{66} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta V \\ \delta\gamma \\ \delta\psi \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \\ B_{41} & B_{42} & B_{43} \\ B_{51} & B_{52} & B_{53} \\ B_{61} & B_{62} & B_{63} \end{bmatrix} \begin{bmatrix} \delta\eta \\ \delta n \\ \delta\sigma \end{bmatrix}. \tag{32}$$

Set the given MAV trajectory as the reference, the state space is defined as follows:

$$\begin{gathered} \delta x = x_W - x_L, \delta y = y_W - y_L, \delta z = z_W - z_L, \\ \delta V = V_W - V_L, \delta\gamma = \gamma_W - \gamma_L, \delta\psi = \psi_W - \psi_L. \end{gathered} \tag{33}$$

The control commands are defined as follows:

$$\delta\eta = \eta - \eta_L, \delta n = n - n_L, \delta\sigma = \sigma - \sigma_L, \tag{34}$$

where A and B are the partial derivative coefficient matrix calculated according to the motion differential equation and the feature points of the reference trajectory. The calculation results are as follows:

$$\begin{aligned}
&A_{11} = A_{12} = A_{13} = 0,\\
&A_{14} = \cos\gamma\sin\psi, A_{15} = -V\sin\gamma\sin\psi, A_{16} = V\cos\gamma\cos\psi,\\
&A_{21} = A_{22} = A_{23} = 0,\\
&A_{24} = \cos\gamma\cos\psi, A_{25} = -V\sin\gamma\cos\psi, A_{26} = -V\cos\gamma\sin\psi,\\
&A_{31} = A_{32} = A_{33} = A_{36} = 0,\\
&A_{34} = \sin\gamma, A_{35} = V\cos\gamma,\\
&A_{41} = A_{42} = A_{46} = 0, A_{43} = D_z/m,\\
&A_{44} = D_V/m, A_{45} = -g\cos\gamma,\\
&A_{51} = A_{52} = A_{53} = A_{56} = 0,\\
&A_{54} = -g(n\cos\sigma - \cos\gamma)/V^2, A_{55} = g\sin\gamma/V,\\
&A_{61} = A_{62} = A_{63} = A_{66} = 0,\\
&A_{64} = g\sin\sigma n/\left(V^2\cos\gamma\right), A_{65} = -gn\sin\sigma\sin\gamma/\left(V\cos^2\gamma\right),\\
&B_{11} = B_{12} = B_{13} = B_{21} = B_{22} = B_{23} = B_{31} = B_{32} = B_{33} = 0,\\
&B_{41} = T_{\max}/m, B_{42} = B_{43} = 0,\\
&B_{51} = -D_n/m, B_{52} = g\cos\sigma/V, B_{53} = -gn\sin\sigma,\\
&B_{61} = 0, B_{62} = -g\sin\sigma/(V\cos\gamma), B_{63} = -gn\cos\sigma/(V\cos\gamma).
\end{aligned} \tag{35}$$

where D_z, D_V and D_n are the partial derivatives of the drag D on the feature point of the reference trajectory to the flight height z, velocity V and normal overload n respectively. Define the optimal control performance index from t_0 to t_f as follows:

$$J = 0.5\int_{t_0}^{t_f}\left[X^{\mathrm{T}}(t)QX(t) + u^{\mathrm{T}}(t)Ru(t)\right]\mathrm{d}t, \tag{36}$$

where Q and R are the weight matrices of state and control respectively. Q is positive semi-definite and R is positive-definite. Then, there exists an optimal control law $u^* = -K^*X$ to minimize the above performance index, and the feedback gain matrix K^* is

$$K^* = \begin{bmatrix} K_{\eta 1} & K_{\eta 2} & K_{\eta 3} & K_{\eta 4} & K_{\eta 5} & K_{\eta 6} \\ K_{n1} & K_{n2} & K_{n3} & K_{n4} & K_{n5} & K_{n6} \\ K_{\sigma 1} & K_{\sigma 2} & K_{\sigma 3} & K_{\sigma 4} & K_{\sigma 5} & K_{\sigma 6} \end{bmatrix}, \tag{37}$$

$$K^* = -R^{-1}B^{\mathrm{T}}P \tag{38}$$

where P is the solution of the Riccati equation. It is calculated by

$$-PA - A^{\mathrm{T}}P + PBR^{-1}B^{\mathrm{T}}P - Q = 0. \tag{39}$$

Define Q and R as follows:

$$\begin{aligned}
&Q = \mathrm{diag}[Q_1, Q_2, Q_3, Q_4, Q_5, Q_6],\\
&R = \mathrm{diag}[R_1, R_2, R_3].
\end{aligned} \tag{40}$$

To reflect the impact of the flight relative distance in the dual aircraft formation flight. Set $Q_1 = Q_2 = Q_3$ and define $\delta D^2 = \Delta D^2 = \delta x^2 + \delta y^2 + \delta z^2$, then

$$\begin{aligned} J = \quad & 0.5\int_{t_0}^{t_f} \left[\left(Q_1 \delta D^2 + Q_4 \delta V^2 + Q_5 \delta \gamma^2 + Q_6 \delta \psi^2 \right) \right. \\ & \left. + \left(R_1 \delta \eta^2 + R_2 \delta n^2 + R_3 \delta \sigma^2 \right) \right] \mathrm{dt}. \end{aligned} \tag{41}$$

According to Bryson Law [38], Q and R are set as follows:

$$\begin{aligned} & Q_1 D^2{}_{\Delta\max} = Q_4 V^2{}_{\Delta\max} = Q_5 \gamma^2{}_{\Delta\max} = Q_6 \psi^2{}_{\Delta\max} \\ & = R_1 \eta^2{}_{\max} = R_2 n^2{}_{\max} = R_3 \sigma^2{}_{\max}. \end{aligned} \tag{42}$$

Set $Q_1 = 1$, then other parameters can be obtained. According to u^*, the control commands can be obtained:

$$\begin{aligned} \eta &= \eta_L - \left(K_{\eta 1} \delta x + K_{\eta 2} \delta y + K_{\eta 3} \delta z + K_{\eta 4} \delta V + K_{\eta 5} \delta \gamma + K_{\eta 6} \delta \psi \right), \\ n &= n_L - \left(K_{n1} \delta x + K_{n2} \delta y + K_{n3} \delta z + K_{n4} \delta V + K_{n5} \delta \gamma + K_{n6} \delta \psi \right), \\ \sigma &= \sigma_L - \left(K_{\sigma 1} \delta x + K_{\sigma 2} \delta y + K_{\sigma 3} \delta z + K_{\sigma 4} \delta V + K_{\sigma 5} \delta \gamma + K_{\sigma 6} \delta \psi \right). \end{aligned} \tag{43}$$

Since the feedback gains obtained at different feature points of the reference trajectory are different, the monotonic flights can be selected as an independent variable, and the feedback gain coefficient of the offline design can be interpolated to obtain the corresponding control commands.

4.3. Experiment of Nominal Conditions

In the experiment of nominal conditions, the initial position of the MAV was $x_{L0} = 0$ m, $y_{L0} = 0$ m, $z_{L0} = 10,000$ m, $V_{L0} = 400$ m/s, $\gamma_{L0} = \pi/6$, $\psi_{L0} = 0$. The initial position of the UAV was $x_{W0} = 100$ m, $y_{W0} = 100$ m, $z_{W0} = 10,000$ m, $V_{W0} = 400$ m/s, $\gamma_{W0} = \pi/6$, $\psi_{W0} = 0$.

The formation flight trajectories of MAV and UAV of the three methods are shown in Figure 5. The MAV is designed to make continuous S-shaped large maneuver with a maximum overload of about 4 g at 1 s, 11 s, 29 s and 41 s, respectively. Figure 5 indicates that the LQR, DDPG, and HIAC can realize the stable tracking of the given trajectory of the MAV under large, overloaded maneuvers and reach the target of the designed formation.

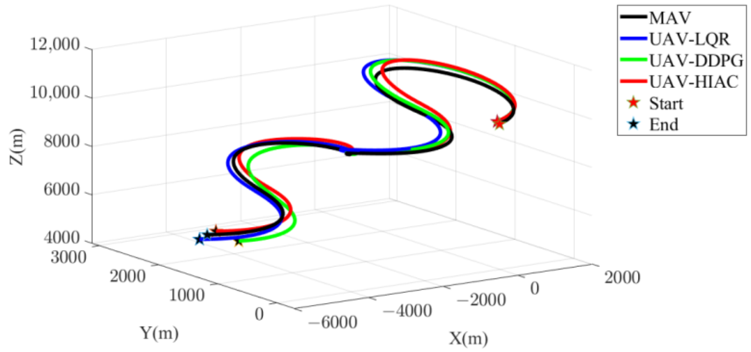
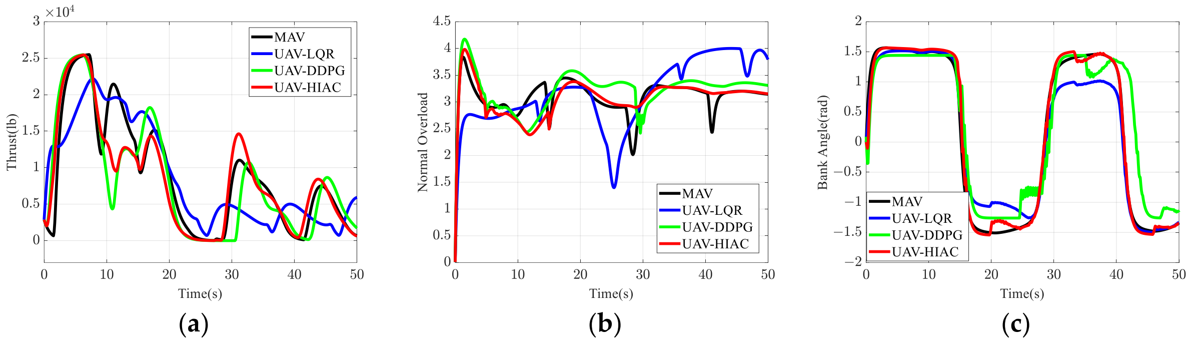


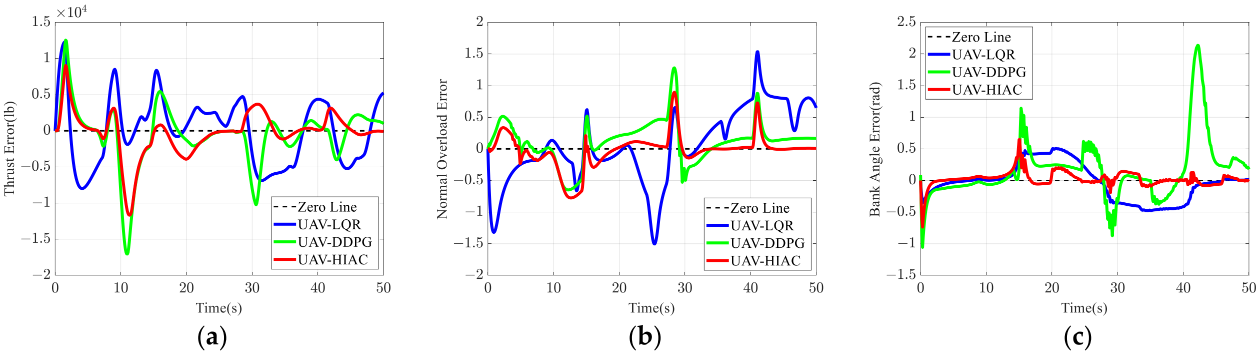
Figure 5. The formation flight trajectory of MAV and UAV of three methods.

Figure 6a–c shows the control commands of the UAV, i.e., the thrust, the normal overload, and the bank angle, generated by the LQR, DDPG and HIAC, respectively, with reference commands of the MAV. Figure 7a–c shows the errors between the control commands of the LQR, DDPG, and HIAC and the reference commands of the MAV. Figure 6 illustrates that there are four peaks in the curves of the control commands due to the four large, overloaded maneuvers. Moreover, compared with the LQR and DDPG, the trend of the control commands of the HIAC can be better consistent with the MAV in thrust, normal overload, and bank angle. Especially in the control of the normal overload, the HIAC has mitigated the sharp change of the commands generated by the reinforcement learning controller to a certain extent. It can provide more smooth and executable control commands under large maneuvers. However, during the large maneuver of the MAV, in order to track the reference commands, it inevitably generates a certain amount of extra adjustment for the thrust, overload, and bank angle for the three methods.



(**a**) (**b**) (**c**)

Figure 6. The control commands of the MAV generated by the LQR, DDPG, and HIAC, respectively, with reference commands of the MAV. The results of the thrust, the thrust, the normal overload, and the bank angle are presented in (**a**–**c**), respectively.



(**a**) (**b**) (**c**)

Figure 7. The errors between the control commands of the LQR, DDPG, HIAC and the reference commands of the MAV. The errors of the thrust, the thrust, the normal overload, and the bank angle are presented in (**a**–**c**), respectively.

Figure 8 shows the change of the three controlled states of the UAV, i.e., the velocity, the flight path angle, and the flight azimuth angle, generated by LQR, DDPG and HIAC. Figure 9 shows the deviation of the three controlled states and the relative distance. It can be seen from Figure 8 that the change trend of the controlled state of the HIAC is basically the same as that of the MAV, and the formation maintenance performance is obviously better than that of the LQR and DDPG. Especially, the HIAC can keep up with most of the fluctuations of the MAV in the flight velocity and the flight azimuth angle. Moreover, Figure 9 shows that compared with the LQR and DDPG, the control precision of the HIAC has been significantly improved, and the control deviation can rapidly decrease to nearly 0 under the large maneuver. Figure 9d indicates that the HIAC successfully limits the formation distance within the safe distance between 100 m and 600 m while LQR and DDPG fail. The LQR continuously accumulates distance deviation due to the velocity deviation during the flight, and ultimately, the formation distance reveals a divergent trend. Meanwhile, although the relative distance of the DDPG gradually converges, it still extends beyond the safe distance at the end of the flight.

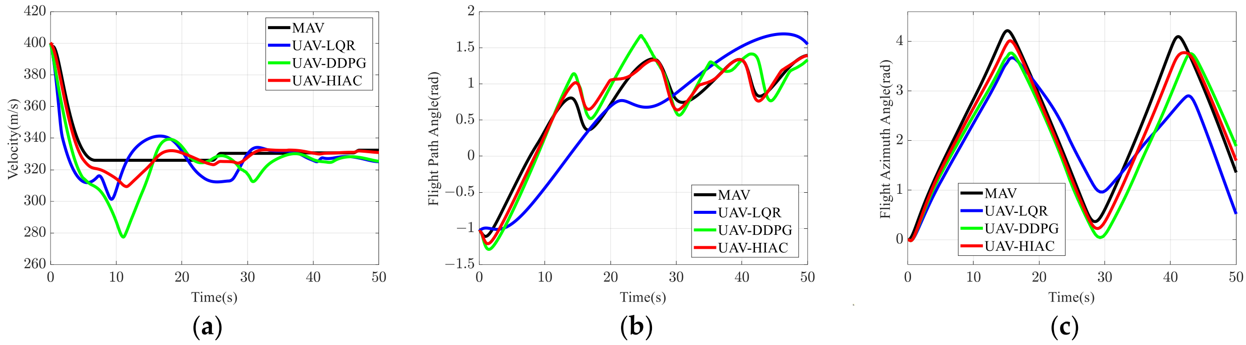


Figure 8. The change of the three controlled state of the UAV generated by the LQR, DDPG and HIAC. The results of the velocity, the flight path angle, and the flight azimuth angle are presented in (**a**–**c**), respectively.

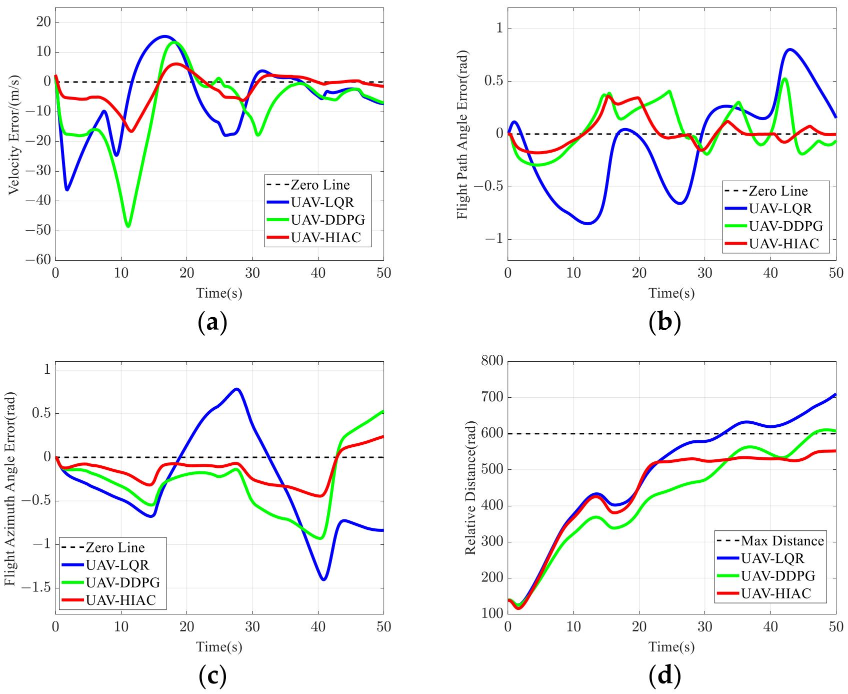


Figure 9. The deviation of the velocity, the flight path angle, the flight azimuth angle and the relative distance are presented in (**a**–**d**), respectively.

Table 3 presents the root mean square (RMS) errors and maximum errors of the four controlled states of the LQR, DDPG and HIAC. It is clear that both the RMS error and maximum error of the HIAC are smaller than those of the LQR and DDPG. Moreover, the HIAC has a reduction of 5.81%, 70.44%, and 64.95%, respectively, in the RMS error of the velocity, flight path angle and flight azimuth angle compared with the LQR, and has a reduction of 60.35%, 55.32% and 69.47% in the maximum error of velocity, flight path angle and flight azimuth angle, respectively, compared with the LQR. The HIAC has a reduction of 36.10%, 35.85% and 51.61%, respectively, in the RMS error of velocity, flight path angle and flight azimuth angle compared with the DDPG, and has a reduction of 54.43%, 31.57% and 55.01% in the maximum error of velocity, flight path angle and flight azimuth angle, respectively, compared with the DDPG.

Table 3. RMS and maximum errors of the four states of the LQR, DDPG, and HIAC in nominal conditions.

Controller		Velocity (m/s)	Flight Path Angle (rad)	Flight Azimuth Angle (rad)	Relative Distance (m) Safe Distance [100, 600]
LQR	RMS	5.6957	0.4737	0.6202	516.7072
	Max.	15.3307	0.8027	0.7833	710.2799
DDPG	RMS	8.3953	0.2183	0.4493	444.1190
	Max.	13.3379	0.5241	0.5315	610.6078
HIAC	RMS	5.3647	0.1401	0.2174	460.0709
	Max.	6.0780	0.3586	0.2391	552.1845

In summary, the proposed HIAC significantly improves the state control performance and guarantees that the flight distance stays within a safe distance as well.

4.4. Monte Carlo Experiments

In order to further test how the HIAC adapts to various initial conditions, 100 Monte Carlo simulations were carried out by adding random deviations to the nominal conditions.

The initial position of the MAV is $x_{L0} = 0$ m, $y_{L0} = 0$ m, $z_{L0} = 10,000$ m, $V_{L0} = 400$ m/s, $\gamma_{L0} = \pi/6$, $\psi_{L0} = 0$. The baseline of initial values of the UAV is $x_{W0} = 100$ m, $y_{W0} = 100$ m, $z_{W0} = 10,000$ m, $V_{W0} = 400$ m/s, $\gamma_{W0} = \pi/6$, $\psi_{W0} = 0$. Then, random deviations which follow the uniform distributions were added to these six baselines, respectively. The specific values of the deviations are presented in Table 4.

Table 4. Uniform distribution of deviations for the six initial values.

Numbers of Monte Carlo Simulations	X (m)	Y (m)	Z (m)	Velocity (m/s)	Flight Path Angle (rad)	Flight Azimuth Angle (rad)
100	[−50, 550]	[−50, 550]	[−1000, 1000]	[−100, 100]	$[-\pi/18, \pi/18]$	$[-\pi/18, \pi/18]$

Figure 10 is the scatterplot of the Monte Carlo simulation results of the velocity errors, flight path angle error, flight azimuth angle, and relative distance for the LQR, DDPG and HIAC. For each evaluation index, the horizontal axis is the RMS error, and the vertical axis is the maximum error. It can be seen that the HIAC can fulfill the control target in the magnitude of velocity. Meanwhile, because the training threshold is set quite strictly in order to achieve better control performance, the maximum error and RMS error of the flight path angle and flight azimuth angle may extend out of the threshold when the extreme deviations are added to the initial values. However, the HIAC can still present a satisfactory control accuracy of the angle compared with the DDPG and LQR. Moreover, in terms of the safety distance, the HIAC can stay within a safe distance of 100 m to 600 m from the MAV, which reaches the distance control target. However, the DDPG and LQR gradually extend out of the safe distance as the initial values vary. Statistically, compared with LQR and DDPG, the HIAC has smaller values in both the RMS error and maximum error of these four evaluation indices. In summary, the performance of the HIAC in formation control is better than that of the other two methods, which is consistent with the simulation results under nominal conditions. It is believed that the HIAC has significant adaptability to the varying initial conditions.

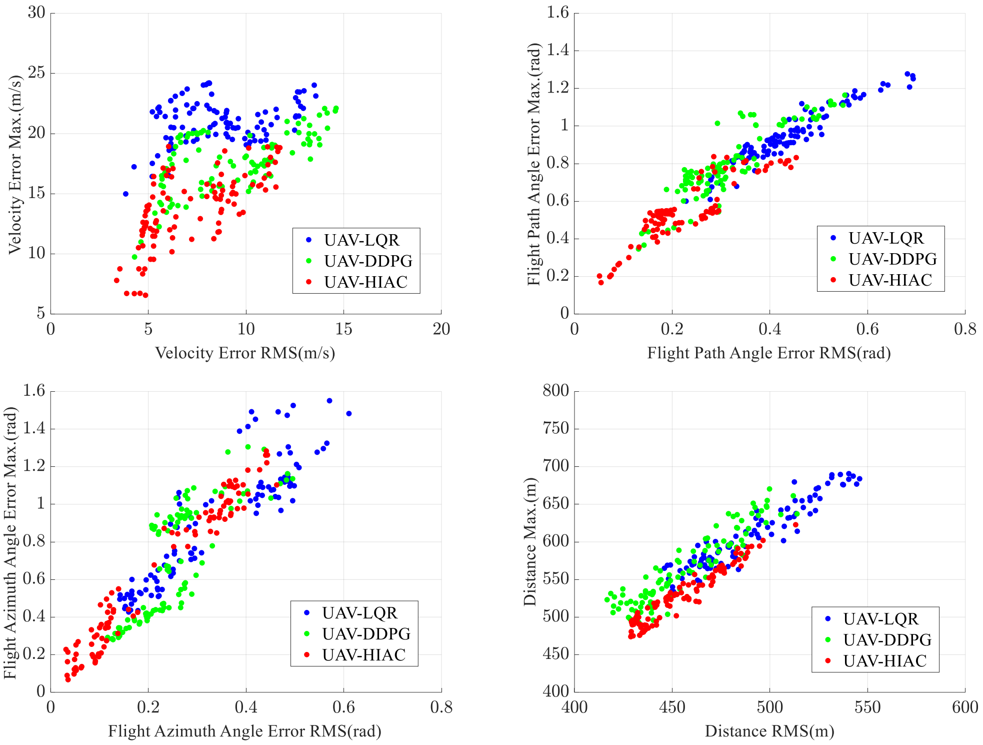


Figure 10. Monte Carlo simulation results of LQR, DDPG and HIAC.

5. Conclusions

In this study, a novel HIAC method was proposed, which is able to enhance the smoothness and executability of control commands and improve the control performance of the MAV/UAV flight formation. First, based on the idea of "meta-action" in hybrid reinforcement learning, the formation control was modeled as a continuous–discrete space control problem. Then, we proposed the framework of the HIAC, and the hybrid intelligent agent model based on the DDPG/DDQN was designed through multi-channel decoupling. Finally, we carried out simulations of nominal conditions and 100 Monte Carlo simulations in varying initial conditions. The simulation results showed that, compared with the traditional LQR and DDPG, the HIAC has better performance of high control precision and rapid convergence. Meanwhile, the adaptability of HIAC to the varying initial conditions was verified as well.

For further practical applications, HIAC can gradually support practical scenarios such as formation military operations and terrain surveys. In particular, two aspects should be considered when applying HIAC. The first is the reliability of the method. HIAC should be preliminarily trained with a large number of ground tests before the real flights, to ensure that intelligent control gradually takes authority over traditional flight-control methods. The second is the portability of the method. At present, the method supports the deployment of reinforcement learning on hardware such as DSP, and FPGA, and can realize airborne portability and the online training of agent models.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, resources, data curation, writing—original draft preparation, L.Z. (Luodi Zhao); writing—review and editing, Y.L. and Q.P.; visualization, investigation, Y.L.; supervision, project administration, funding acquisition, L.Z (Long Zhao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China, grant number 42274037 and 41874034, the National key research and development program of China, grant number 2020YFB0505804 and the Beijing Natural Science Foundation, grant number 4202041.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lei, L.; Wang, T.; Jiang, Q. Key Technology Develop Trends of Unmanned Systems Viewed from Unmanned Systems Integrated Roadmap 2017–2042. *Unmanned Syst. Technol.* **2018**, *1*, 79–84.
2. Mishory, J. DARPA Solicits Information for New Lifelong Machine Learning Program. *Inside Pentagon* **2017**, *33*, 10.
3. Pittaway, N. Loyal Wingman. *Air Int.* **2019**, *96*, 12–13.
4. Oh, K.; Park, M.; Ahn, H. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [CrossRef]
5. Wang, H.; Liu, S.; Lv, M.; Zhang, B. Two-Level Hierarchical-Interaction-Based Group Formation Control for MAV/UAVs. *Aerospace* **2022**, *9*, 510. [CrossRef]
6. Choi, I.S.; Choi, J.S. Leader-Follower formation control using PID controller. In Proceedings of the International Conference on Intelligent Robotics & Applications, Montreal, QC, Canada, 3–5 October 2012.
7. Gong, Z.; Zhou, Z.; Wang, Z.; Lv, Q.; Xu, Q.; Jiang, Y. Coordinated Formation Guidance Law for Fixed-Wing UAVs Based on Missile Parallel Approach Metho. *Aerospace* **2022**, *9*, 272. [CrossRef]
8. Liang, Z.; Ren, Z.; Shao, X. Decoupling trajectory tracking for gliding reentry vehicles. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 115–120.
9. Kuriki, Y.; Namerikawa, T. Formation Control of UAVs with a Fourth-Order Flight Dynamics. *J. Control. Meas. Syst. Integr.* **2014**, *7*, 74–81. [CrossRef]
10. Kuriki, Y.; Namerikawa, T. Consensus-based cooperative formation control with collision avoidance for a multi-UAV system. In Proceedings of the American Control Conference, Portland, OR, USA, 4–6 June 2014.
11. Atn, G.M.; Stipanovi, D.M.; Voulgaris, P.G. Collision-free trajectory tracking while preserving connectivity in unicycle multi-agent systems. In Proceedings of the American Control Conference, Washington, DC, USA, 17–19 June 2013.
12. Tsankova, D.D.; Isapov, N. Potential field-based formation control in trajectory tracking and obstacle avoidance tasks. In Proceedings of the Intelligent Systems, Sofia, Bulgaria, 6–8 September 2012.
13. Hu, J.; Wang, L.; Hu, T. Autonomous Maneuver Decision Making of Dual-UAV Cooperative Air Combat Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 467. [CrossRef]
14. Luo, Y.; Meng, G. Research on UAV Maneuver Decision-making Method Based on Markov Network. *J. Syst. Simul.* **2017**, *29*, 106–112.
15. Yang, Q.; Zhang, J.; Shi, G. Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 363–378. [CrossRef]
16. Li, Y.; Han, W.; Wang, Y. Deep Reinforcement Learning with Application to Air Confrontation Intelligent Decision-Making of Manned/Unmanned Aerial Vehicle Cooperative System. *IEEE Access* **2020**, *99*, 67887–67898. [CrossRef]
17. Wang, X.; Gu, Y.; Cheng, Y. Approximate Policy-Based Accelerated Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 1820–1830. [CrossRef] [PubMed]
18. Hasselt, H.V.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Canberra, Australia, 30 November–5 December2015.
19. Mnih, V.; Kavukcuoglu, K.; Silver, D. Playing Atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
20. Silver, D.; Huang, A.; Maddison, C.J. Mastering the game of go with deep neural networks and the tree search. *Nature* **2016**, *529*, 484. [CrossRef]
21. Silver, D.; Lever, G.; Heess, N. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
22. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
23. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
24. Wada, D.; Araujo-Estrada, S.A.; Windsor, S. Unmanned Aerial Vehicle Pitch Control Using Deep Reinforcement Learning with Discrete Actions in Wind Tunnel Test. *Aerospace* **2021**, *8*, 18. [CrossRef]
25. Haarnoja, T.; Zhou, A.; Abbeel, P. Soft Actor-Critic Algorithms and Applications. *arXiv* **2018**, arXiv:1812.05905.
26. Heess, N.; Silver, D.; Teh, Y.W. Actor-critic reinforcement learning with energy-based policies. In Proceedings of the Tenth European Workshop on Reinforcement Learning, Edinburgh, UK, 30 June–1 July 2012.
27. Schaul, T.; Quan, J.; Antonoglou, I. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
28. Hu, Z.; Wan, K.; Gao, X.; Zhai, Y.; Wang, Q. Deep Reinforcement Learning Approach with Multiple Experience Pools for UAV's Autonomous Motion Planning in Complex Unknown Environments. *Sensors* **2020**, *20*, 1890. [CrossRef] [PubMed]
29. Neunert, M.; Abdolmaleki, A.; Wulfmeier, M. Continuous-Discrete Reinforcement Learning for Hybrid Control in Robotics. In Proceedings of the Conference on Robot Learning, Virtual Event, 30 October–1 November 2020.
30. Xiong, J.; Wang, Q.; Yang, Z. Parametrized Deep Q-Networks Learning: Reinforcement Learning with Discrete-Continuous Hybrid Action Space. *arXiv* **2018**, arXiv:1810.06394.
31. Anderson, M.R.; Robbins, A.C. Formation flight as a cooperative game. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Boston, MA, USA, 10–12 August 1998.

32. Kelley, H.J. Reduced-order modeling in aircraft mission analysis. *AIAA J.* **2015**, *9*, 349–350. [CrossRef]
33. Williams, P. Real-time computation of optimal three-dimensional aircraft trajectories including terrain-following. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 24–26 August 2006.
34. Wang, X.; Guo, J.; Tang, S. Entry trajectory planning with terminal full states constraints and multiple geographic constraints. *Aerosp. Sci. Technol.* **2019**, *84*, 620–631. [CrossRef]
35. Snell, S.A.; Enns, D.F.; Garrard, W.L. Nonlinear inversion flight control for a supermaneuverable aircraft. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, USA, 20–22 August 1990.
36. Dukeman, G. Profile-Following Entry Guidance Using Linear Quadratic Regulator Theory. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, USA, 5–8 August 2002.
37. Wen, Z.; Shu, T.; Hong, C. A simple reentry trajectory generation and tracking scheme for common aero vehicle. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012.
38. Bryson, A.E.; Ho, Y. Applied Optimal Control. *Technometrics* **1979**, *21*, 3.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.