sensors

MDPI

Article

Delay-Informed Intelligent Formation Control for UAV-Assisted IoT Application

Lihan Liu [1], Mengjiao Xu [2], Zhuwei Wang [2,*], Chao Fang [2,3], Zhensong Li [4], Meng Li [2], Yang Sun [2] and Huamin Chen [2]

[1] School of Statistics and Data Science, Beijing Wuzi University, Beijing 101149, China; liulihan@bwu.edu.cn
[2] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; xumengjiao@emails.bjut.edu.cn (M.X.); fangchao@bjut.edu.cn (C.F.); limeng720@bjut.edu.cn (M.L.); sunyang@bjut.edu.cn (Y.S.); chenhuamin@bjut.edu.cn (H.C.)
[3] Purple Mountain Laboratory: Networking, Communications and Security, Nanjing 210096, China
[4] School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China; lizhensong@bistu.edu.cn
* Correspondence: wangzhuwei@bjut.edu.cn; Tel.: +86-188-1140-5575

Abstract: Multiple unmanned aerial vehicles (UAVs) have a greater potential to be widely used in UAV-assisted IoT applications. UAV formation, as an effective way to improve surveillance and security, has been extensively of concern. The leader–follower approach is efficient for UAV formation, as the whole formation system needs to find only the leader's trajectory. This paper studies the leader–follower surveillance system. Owing to different scenarios and assignments, the leading velocity is dynamic. The inevitable communication time delays resulting from information sending, communicating and receiving process bring challenges in the design of real-time UAV formation control. In this paper, the design of UAV formation tracking based on deep reinforcement learning (DRL) is investigated for high mobility scenarios in the presence of communication delay. To be more specific, the optimization UAV formation problem is firstly formulated to be a state error minimization problem by using the quadratic cost function when the communication delay is considered. Then, the delay-informed Markov decision process (DIMDP) is developed by including the previous actions in order to compensate the performance degradation induced by the time delay. Subsequently, an extended-delay informed deep deterministic policy gradient (DIDDPG) algorithm is proposed. Finally, some issues, such as computational complexity analysis and the effect of the time delay are discussed, and then the proposed intelligent algorithm is further extended to the arbitrary communication delay case. Numerical experiments demonstrate that the proposed DIDDPG algorithm can significantly alleviate the performance degradation caused by time delays.

Keywords: surveillance; formation control; intelligent control strategy; time delay; dynamic leading velocity

Citation: Liu, L.; Xu, M.; Wang, Z.; Fang, C.; Li, Z.; Li, M.; Sun, Y.; Chen, H. Delay-Informed Intelligent Formation Control for UAV-Assisted IoT Application. *Sensors* **2023**, *23*, 6190. https://doi.org/10.3390/s23136190

Academic Editor: Petros S. Bithas

Received: 22 May 2023
Revised: 29 June 2023
Accepted: 4 July 2023
Published: 6 July 2023

1. Introduction

Recently, the development of unmanned aerial vehicles (UAVs) has brought many benefits in UAV-assisted application fields, such as surveillance, rescue, reconnaissance and search [1,2]. The UAV formation control, driving each vehicle to reach the prescribed constraint on its own states through generating appropriate control commands, significantly expands the potential applications and opens up new possibilities for UAVs. For example, a group of UAVs could expand the fields of view when executing assignment.

A task of cooperative surveillance is considered in this paper. The target is to guide a group of UAVs equipped with cameras to fly over an urban area (possibly hostile) to provide complete surveillance coverage in an optimal manner [3]. Considering the limitation of batteries, leader–follower units are introduced to make a group of UAVs fly with a formation in order to improve the efficiency and expand the field of surveillance.

In this paper, the leader is assigned to make the flying strategy, like the flying velocity and trajectory position, depend on the environment information transmitted by wireless sensor agent networks (WSANs). Followers focus on tracking with a dynamic leader and keep a desired cooperation formation. In this paper, we focus on the design of the controller to make followers achieve the desired cooperation formation while tracking a dynamic leader.

However, UAVs are underactuated systems constrained by high mobility and serious disturbances [4]. Therefore, it becomes a great challenge to address the robust formation controller design problem to enable UAVs to achieve the desired cooperation formation. The traditional optimal formation control methods, such as the nonlinear model predictive control (see [5,6]), and nonlinear PID control (see [7,8]), are proposed to alleviate the degradation of control stability attributed to the external disturbances and uncertainties in the UAV formation. These approaches can generally be regarded as a cost function minimization problem defined by a set of UAV states and control actions. Unfortunately, the above methods often fail to generalize to the wider range of application scenarios due to the highly dynamic and time-varying features of UAVs.

Existing approaches have been proposed to overcome the limitations of traditional formation control algorithms, among which the highest potential one is reinforcement learning (RL) [9]. In fact, RL is a classical learning method to address the sequential decision-making problem within the Markov decision process (MDP). At each step, the agent interacts with the environment and derives a reward. After exploration and training, the control policy gradually achieves the optimal tratego. By using the framework of MDP, RL is a typical algorithm developed in the control field originally for optimal stochastic control under uncertainty [10]. Different from the classical rule-based optimization methods, RL learns intelligently in each step, interacting with the environment to derive approximate optimal model parameters.

In order to improve the learning ability of RL, deep reinforcement learning (DRL), integrating the benefits of both RL and deep neural networks (DNNs), has been proposed. DRL can efficiently handle a much more complicated state space and dynamic environment, and achieve superior performance for game-playing tasks [11–13]. DRL has become a research hotspot in the field of UAV control, such as the outer-loop control (formation maintenance, navigation [14], path planning [15]) and inner-loop control (altitude [16]). In DRL, the deep Q learning (DQN) technique is employed to reduce the correlation among successive experience samples by using an experience replay buffer. Nevertheless, DQN can only deal with a limited action space, while the UAV formation control is a continuous control process with an unlimited action space. Then, the actor–critic method is further developed for continuous control action [9]. Based on the actor–critic framework, the deep deterministic policy gradient (DDPG) algorithm, which takes advantage of the DQN experience replay and dual network structure to enhance the deterministic policy gradient (DPG) algorithm, has been used comprehensively for continuous agent control, and its feasibility has been validated in many potential scenarios, such as autonomous driving, (longitudinal see [17], mixed-autonomy see [18]), UAV (navigation see [19], motion control see [20]), etc.

Formation control requires continual and real-time information exchange. At each time interval, environment information should be exchanged (i.e., sent or received) by sensor nodes through the WSANs, which typically suffers from a series of issues, such as network topology, network traffic and system resource limitations, resulting in inevitable network-induced time delays. In our surveillance study, the leader collect the environment information through the sensor nodes spread in WSANs to make the flying strategy, including velocity and position. Then, the new flying strategy, including the velocity and position of the leader, is subsequently transmitted to the follower.

Considering the leader–follower units as an whole unit, this whole unit collects environment information through WSANs and produces action, like an agent in MDP. Consequently, the agent's observations of its environment are not immediately available due to the quality of WSANs, and the time delay actually exists in the action selection and

actuation of the agent in MDP. However, most existing DRL-based algorithm designs are restrained to synchronous systems with delay-free observations and action actuation [21–23]. Therefore, it is of great practical significance to investigate the intelligent UAV formation control considering the time delay constraint. In this paper, we propose a novel intelligent formation control algorithm to deal with the time delay issue in accordance with the model-based DDPG.

1.1. Related Works

The UAV formation control includes three typical types, such as formation generation and maintenance, formation shape maintenance and regeneration and formation maintenance while trajectory tracking [24]. Refs. [25–29] integrate these types into an optimal formation tracking problem. Although these works have the capability to meet the formation maintenance requirement, they fail to deal with much more complex environments because the algorithm parameters cannot be intelligently adjusted according to the dynamic feature of environments. Therefore, it is meaningful to introduce RL algorithms to UAV formation control.

Several new techniques are developed based on the DRL to address the UAV control problem. The DQN algorithm is employed in [30] for real-time UAV path planning. A double deep Q-network (DDQN) is further trained in [15] using the experience replay buffer in order to learn to generate the control policy according to time-varying scenario parameters for UAV. Li et al. [14] focus on the ground target tracking to solve the obstacle problem for UAV system using the improved DDPG. In [31], an end-to-end DRL model is developed for the indoor UAV target searching. Unfortunately, the research of DRL-based UAV formation maintenance is still not enough. In addition, these studies have ignored the effect of the time delay issue, which is an inherent feature in actual UAV formation.

Currently, the study of the RL-based algorithm design with delays is attracting more and more attention. For example, in the design of MDP, Walsh et al. [32] first directly increased the length of a sampling interval in order to achieve the agent's action synchronization using the delayed observations, and then the authors further introduced the delayed actions to the state, which effectively compensates for the effect of time delay. Refs. [33–38] formally described the concept of delayed MDP, and demonstrated that the delayed MDP can be transformed into an equivalent standard MDP, and then it can be employed to formulate the delay-resolved RL framework to derive the near-optimal rewards interacting with the environments. In [39], a delay-aware MDP is proposed to address the continuous control task by increasing the state space with a sequence being executed in the next delay duration step. The interaction manner they proposed is motivated by applying an action buffer as an interval. The agent can obtain environment observation as well as the future sequences from the action buffer, and then determine its future action.

In general, the above methods can be divided into two types, one is that the state space of the learning agent is integrated with the delayed action, and the other is to learn a model of the underlying delay-free process to predict the control actions for future states. Motivated by the existing RL approaches with time delays, the design of UAV formation tracking based on deep reinforcement learning is further developed in our work to address the UAV formation problem in the presence of time delays. In fact, there are few works to address the influence of time delays on intelligent UAV formation in highly dynamic scenarios. However, considering the actual real-time formation control, the time delay is an inherent feature that needs to be studied to improve the control stability.

1.2. Contribution

Due to the uncertainty of wireless communications, the information transmission in UAV formation control will suffer from time delays, which may lead to control instability and formation performance degradation, especially in the high dynamic applications [40–42]. Neither different from the intelligent algorithm in [43], which ignores the influence of time delay, nor different from traditional control methods, such as Artstein's

model reduction [44] and Smith predictor [45], which are restrained to be applied to much more complex and dynamic scenarios because of their limited intelligent adaptability, a delay-informed intelligent framework is proposed in the paper to address the UAV formation problem subject to time delays. The main contributions of our work are as follows:

- In order to regulate the UAV motion, the UAV formation model considering time delay is first established in discrete-time form based on the UAV error dynamics. Then, an optimization problem designed to minimize the quadratic cost function is formulated for the optimal formation control under time delays.
- According to the error dynamics and optimization formation control problem, a delay-informed MDP (DIMDP) framework is presented by including the previous control actions into the state and reward function. Then, a DRL-based algorithm is proposed to address DIMDP, and the classical DDPG algorithm is extended as a delay-informed DDPG (DIDDPG) algorithm to solve DIMDP.
- The computational complexity analysis and the effect of the time delay are discussed, and the proposed algorithm is further extended to the arbitrary communication delay case. Through the training results, the proposed DIDDPG for the UAV formation control can achieve better convergence and system performance.

The rest of this paper is organized as follows. The system model and UAV formation optimization problem are presented in Section 2. In Section 3, the environment model is established as DIMDP, and then the DIDDPG algorithm is proposed to solve DIMDP. Section 4 shows the simulation results, and Section 5 concludes our work.

2. System Modeling and Problem Formulation

In this section, by considering the time delay and dynamic leader velocity, the formation control model is first presented. Then, the cost function based on the discrete-time states errors is designed for the follower to reach the desired states. Finally, the optimization problem is formulated.

2.1. System Modeling

UAV formation can be applied to a multitude of security and surveillance areas. The pattern formation is crucial for multi-UAV formation control mechanisms while cautiously navigating the surveillance areas. The leader–follower formation is introduced to improve the efficiency for UAV formation, as the surveillance system needs to find only the leader's trajectory.

In this paper, the UAV formation is divided into several leader–follower control units, with one UAV designated as the leader and the remaining UAVs are as followers. By realizing the tracking mission of each unit, the mission of the whole formation is realized. In the formation control process, wireless communication technology is used to complete the information collection and sharing through the WSAN. The leader can receive mission and formation information, and then use the received information to plan the trajectory and guide the direction of the entire formation. The controller regularly collects the position, speed and other status information of the leader and the follower, and calculates the state error of the follower, and then generates and transmits the control strategy to the follower actuator to ensure the stability of formation control. At the same time, communication delays, including leader-to-controller, controller-to-follower actuator, and information processing delays are introduced.

The considered formation control model and corresponding timing diagram are shown as Figures 1 and 2, respectively. The leader is assigned to make the flying strategy, like the flying trajectory and speed, depend on the shared environment information, such as mission and formation information transmitted through the WSAN. The leader makes an appropriate strategy, such as acceleration, deceleration and hover, due to the relevant real-world scenarios and assignments. For example, the formation needs to change when encountering obstacles. Then, the updated formation state information is transmitted to the controllers through the WSAN. Therefore, the leader-to-controller delay is introduced.

Once the formation information is collected, the controller can calculate and generate the control strategy, and then transmit it to the follower actuator to improve the formation control. Meanwhile, the controller-to-follower actuator delay and data processing delay are introduced.

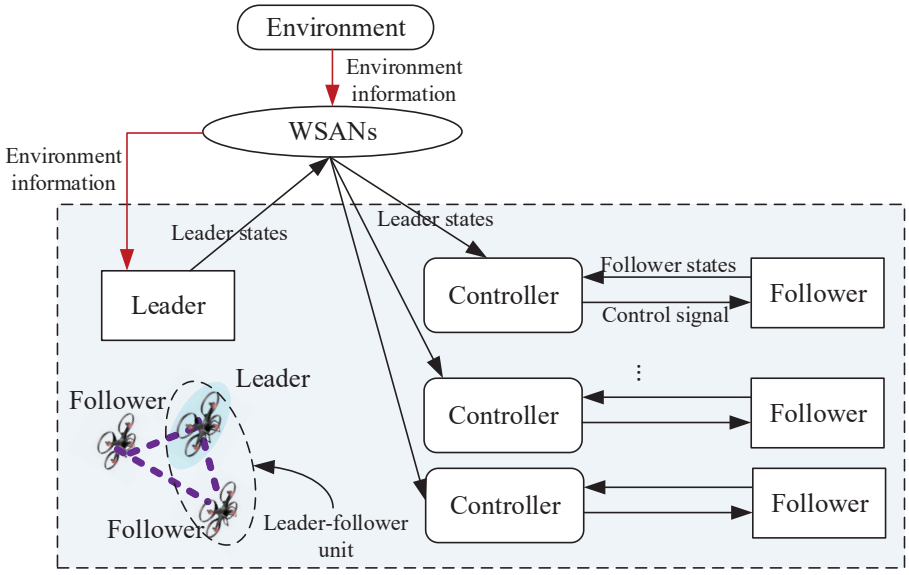

Figure 1. UAV formation system model.

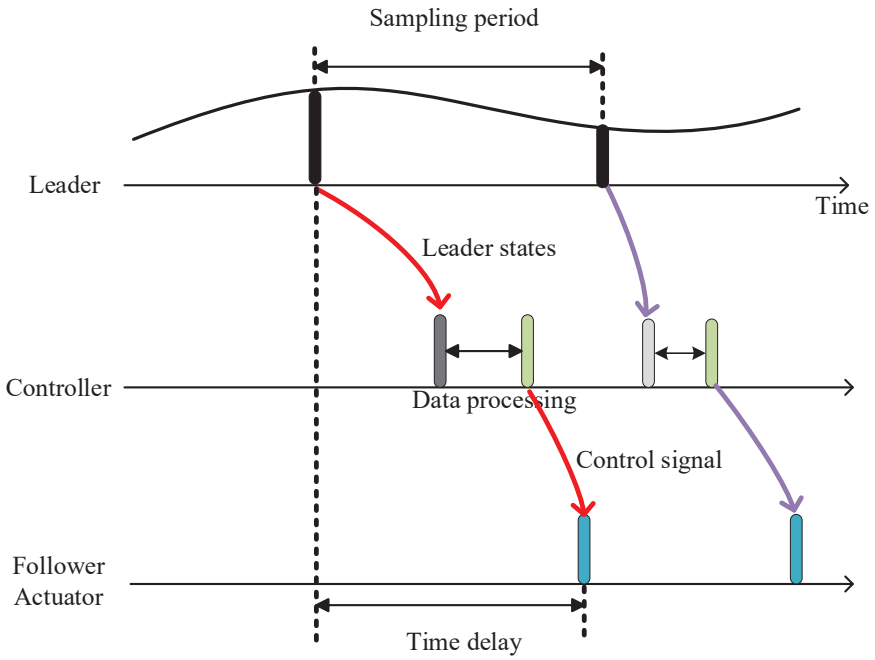

Figure 2. Timing diagram for the leader–follower formation control.

In fact, the location of controller can be placed on the leader UAV or the follower UAV or the ground control center according to the real-world application scenarios. For example, in [21], an intelligent controller placed in the follower is proposed and it is testified that this approach is applicable in many applications, such as penetration and remote surveillance. Figure 2 is able to include all the delay cases, no matter where the controller is placed; due to this, Figure 2 shows a general case for communication delays of formation control. For example, when the controller is placed on the follower, the time delay from the controller to the follower actuator will be small or even negligible. In our work, the dy-

namic leading velocity and time delay are considered due to the complex environment and real-world application.

Considering a leader–follower unit, the kinematics of the follower is given by

$$\begin{aligned} \dot{p}(t) &= v(t), \\ \dot{v}(t) &= c(t - \tau(t)), \end{aligned} \tag{1}$$

where $v(t)$ and $p(t)$ are the velocity and position of the follower, respectively, $c(t)$ denotes the acceleration of the follower (i.e., the control strategy), $\tau(t)$ is the time delay shown as in Figure 2, which accounts to the signal processing delay and the transmission latency from the leader to the controller and from the controller to the follower, and the time delay is typically assumed to be stochastic due to the quality of WSANs.

The model of desired states can be described as [46]

$$\begin{aligned} \dot{p}_r(t) &= v_r(t), \\ \dot{v}_r(t) &= f_r(p_r(t), v_r(t)), \end{aligned} \tag{2}$$

where $v_r(t)$ and $p_r(t)$ are the expected velocity and position, respectively, which are determined by the state of the leader, and $f_r(p_r(t), v_r(t))$ denotes the time-varying acceleration of the leader.

The objective of the follower is to maintain the formation and track the leader. Define the state errors of the follower as follows:

$$\begin{aligned} \Delta p(t) &= p(t) - p_r(t), \\ \Delta v(t) &= v(t) - v_r(t). \end{aligned} \tag{3}$$

Then, based on Formulas (1)–(3) , the relationship among state errors can be deduced as

$$\begin{aligned} \Delta\dot{p}(t) &= \dot{p}(t) - \dot{p}_r(t) = v(t) - v_r(t) = \Delta v(t), \\ \Delta\dot{v}(t) &= \dot{v}(t) - \dot{v}_r(t) = c(t - \tau(t)) - f_r(p_r(t), v_r(t)). \end{aligned} \tag{4}$$

which indicates that the differential of the position error presents the change in velocity, and the differential of the velocity error denotes the change in acceleration.

Note that $\tau(t)$ is a time-varying item due to the uncertainty of the transmission environment, and $f_r(p_r(t), v_r(t))$ is an unknown item due to the dynamic feature of the leader acceleration.

2.2. Optimization Problem Formulation

Define $z(t) = [\Delta p^x(t),\ \Delta v^x(t),\ \Delta p^y(t),\ \Delta v^y(t),\ \Delta p^z(t),\ \Delta v^z(t)]^T$ as the state vector, where the superscripts x, y and z represent the 3D information of state errors. Based on the stat error model (4), the follower dynamics can be expressed as follows:

$$\dot{z}(t) = Az(t) + B[c(t - \tau(t)) - f_r(p_r(t), v_r(t))], \tag{5}$$

where

$$A = \begin{bmatrix} \bar{A} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \bar{A} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & \bar{A} \end{bmatrix},$$

$$B = \begin{bmatrix} \bar{B} & 0_{2\times1} & 0_{2\times1} \\ 0_{2\times1} & \bar{B} & 0_{2\times1} \\ 0_{2\times1} & 0_{2\times1} & \bar{B} \end{bmatrix},$$

that $0_{i\times j}$ is the zero matrix, and

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \bar{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

During each sampling interval, the controller receives the measurement state information, and then derived the control strategy to improve the formation control stability. Then, the corresponding discrete-time dynamics of the follower in the j-th sampling interval $[jT, (j+1)T)$ is given by

$$z_{j+1} = Ez_j + D_j^1 c_j + D_j^2 c_{j-1} + G_j, \tag{6}$$

where

$$E{=}e^{AT}, \ D_j^1 = \int_0^{T-\tau_j} e^{At}dtB, \ D_j^2 = \int_{T-\tau_j}^{T} e^{At}dtB,$$
$$G_j = \int_{jT}^{(j+1)T} e^{A[(j+1)T-s]} f_r(p_r(s), v_r(s))dsB,$$

and z_j and τ_j are the sampled values of $z(t)$ and $\tau(t)$ at time jT, respectively, and c_j denotes the control signal relevant to the received state z_j.

Note that the time delay τ_j causes the time-varying feature of D_j^1 and D_j^2, and the dynamic leader movement also introduces the uncertain item G_j, which increases the difficulty for traditional algorithms to address these dynamic features. Additionally, in each sampling interval, the influence of the previous control signals is further introduced due to the time delays.

The objective of the follower is to minimize state errors. Therefore, the typical quadratic optimization problem for formation control can be formulated as [22]

$$\begin{aligned} &\min_{\{c_j\}} \mathbb{E}\left[z_N^T P z_N + \sum_{j=0}^{N-1} (z_j^T P z_j + c_j^T Q c_j) \right] \\ &s.t. \ z_{j+1} = Ez_j + D_j^1 c_j + D_j^2 c_{j-1} + G_j, \end{aligned} \tag{7}$$

where $\mathbb{E}$ denotes the expectation based on the stochastic natures of the leader movement and time delays, P and Q are system parameters, and N is the finite time horizon.

3. DIDDPG Algorithm for Formation Control

In this section, the DIMDP framework is first presented, and then the environment model which maps the system model to the interaction environment of DIMDP is formulated. Additionally, a DIDDPG algorithm for the UAV formation controller design is proposed.

3.1. DIMDP-Based Environmental Model

The framework of MDP for the leader–follower formulation is shown as in Figure 3. At each time slot, based on the observed current UAV states from the environment, the action is generated and executed according to the action policy. Then, the new state is updated by the state transition function, and the corresponding reward is returned to the agent. In the framework of MDP, the actor0-critic structure, integrating the advantages of the policy search method with the value function learn, is used.

Considering the fact that time delay is an inevitably negative factor to the real-time control, in order to address the optimization formation problem in (7), the MDP framework associated with the time delay needs to be formulated. In fact, the basic MDP framework typically assumes that the system's current states are always available to the agent and the agent always takes relevant actions immediately. However, these assumptions are not

appropriate for the optimization formulation problem because of the time delay. How to integrate the effect of time delay into the MDP framework design is the key issue. Therefore, DIMDP, the standard MDP extension with time delay, is proposed, in which the agent interacts with the environment, and the environment is influenced by the delayed control strategies (i.e., the delayed actions). Below, the detailed definitions of the state space, action space, state transition function and reward function for the DIMDP are given.

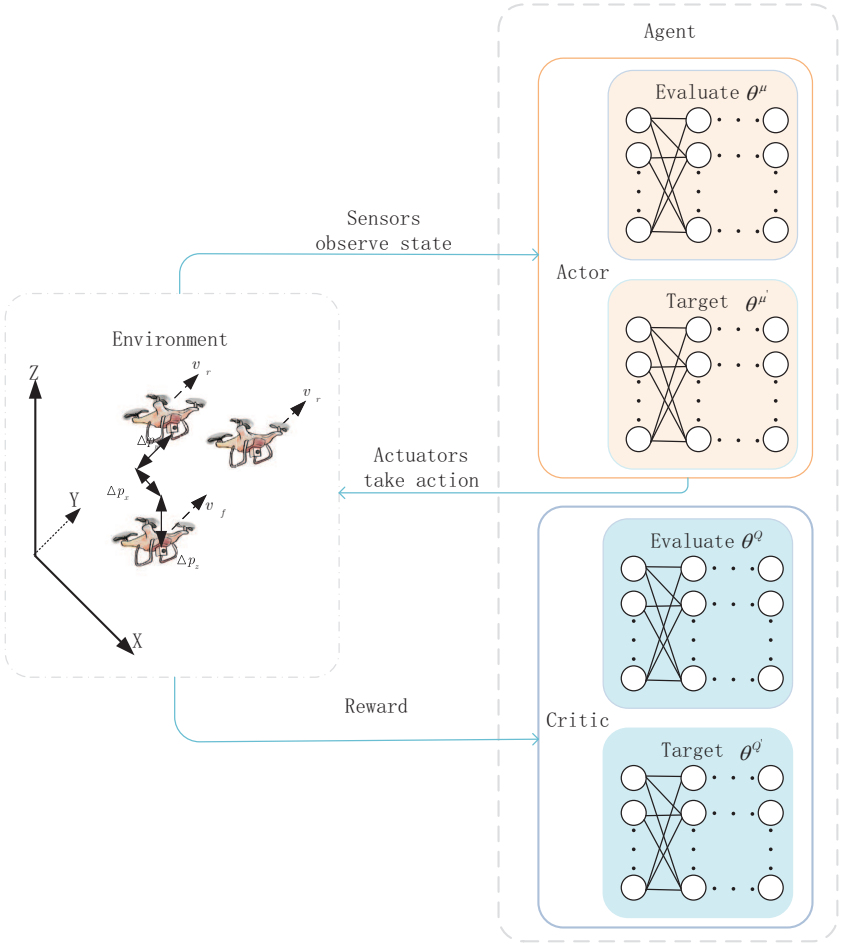

Figure 3. The framework of MDP for the leader–follower formulation.

(1) State: Referring to the leader–follower UAV formation, several factors, including the action of the follower and the error states between the leader and the follower, are considered. As shown in (4), the state errors of the follower are determined by the position and velocity errors. From the discrete-time dynamics (6), the effect of the previous control strategy c_{j-1} is also attributed to the time delay as shown in Figure 4. Therefore, the state in the *j-th* sampling interval is defined as

$$\begin{aligned} s_j &= \left[{z_j}^T,\ {c_{j-1}}^T\right] \\ &= \left[\Delta p_j^x,\ \Delta p_j^y,\ \Delta p_j^z,\ \Delta v_j^x,\ \Delta v_j^y,\ \Delta v_j^z,\ \Delta a_{j-1}^x,\ \Delta a_{j-1}^y,\ \Delta a_{j-1}^z\right]^T. \end{aligned} \tag{8}$$

In (8), the updated state error information and local previous control strategy information are extracted to represent the environment state to regulate the follower UAV tracking. In particular, the previous control strategy is used to compensate for the effects of the time delay.

(2) Action: The decision action is given by

$$a_j = c_j^T, \tag{9}$$

where a_j is actually the acceleration policy of the follower UAV, which is a continuous value, and we have

$$c_{\min} \leq a_j \leq c_{\max}, \tag{10}$$

which indicates that the action is constrained by boundary values.

(3) State transition function: The state transition function can be determined according to the discrete-time dynamics of the follower in (6) as follows:

$$s_{j+1} = s_j F_j + a_j H_j + \left[G_j^T, 0_{3\times 1}\right], \tag{11}$$

where

$$F_j = \begin{bmatrix} E & 0_{3\times 6} \\ D_j^2 & 0_{3\times 3} \end{bmatrix}, \; H_j = \begin{bmatrix} D_j^1 \\ I_{3\times 3} \end{bmatrix}.$$

(4) Reward function: The reward is used to evaluate the performance of the action, and then the follower can intelligently learn to derive the proper control strategy to maintain the formation tracking. The reward function can be designed as the opposite of the cost function in terms of the optimization problem in (7) as follows:

$$r_j = -s_j \bar{P} s_j^T - a_{j+1} Q a_{j+1}^T, \tag{12}$$

where

$$\bar{P} = \begin{bmatrix} P & 0_{3\times 6} \\ 0_{6\times 3} & 0_{3\times 3} \end{bmatrix}.$$

In fact, the closer the follower's states are to the desired ones, the greater the reward. It is significant that, based on the well-designed reward function, the follower can rapidly achieve the desired position and velocity by continuously adjusting the action in order to acquire the maximum long-term cumulative rewards, which is formulated as a finite horizon N item by

$$G_N = -\sum_{j=0}^{N-1} \gamma^j \left(s_j \bar{P} s_j^T + a_j Q a_j^T\right), \tag{13}$$

where γ is a discount factor.

3.2. DIDDPG UAV Formation Algorithm

In this section, we employ the DDPG method with the DIMDP definitions, and then a model-based DIDDPG algorithm for the continuous UAV formation control is proposed.

The framework of DIDDPG is presented as in Figure 5. The main network includes two parts (i.e., critic network and actor network). The actor network $\mu(s|\theta^\mu)$ builds a mapping from states to actions, and the main policy is generated, while the critic network $Q\left(s, a|\theta^Q\right)$ estimates the action value, where θ^Q and θ^μ are parameters of the critic network and actor network, respectively. The target network is employed for the actor–critic architecture to acquire a stable target Q value. The parameters of target network $\mu'(s|\theta^{\mu'})$ and target critic $Q'\left(s_{j+1}, a'|\theta^{Q'}\right)$ update based on the main network parameters.

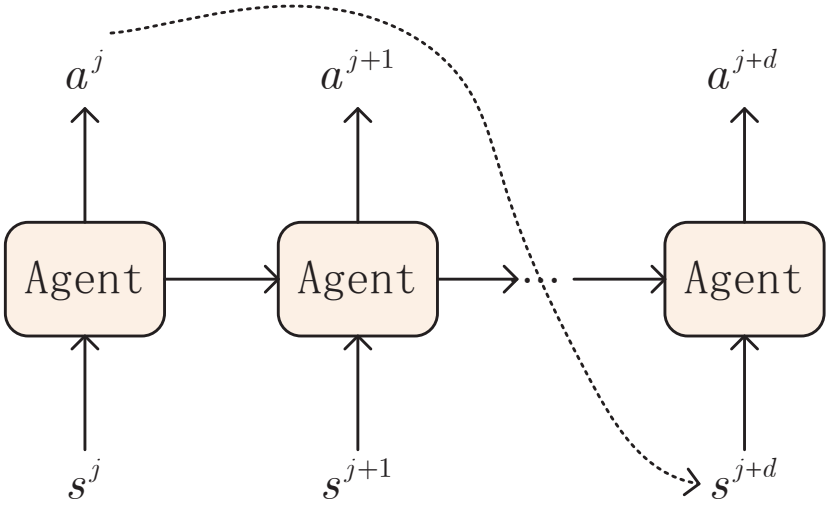

Figure 4. Delay-informed MDP.

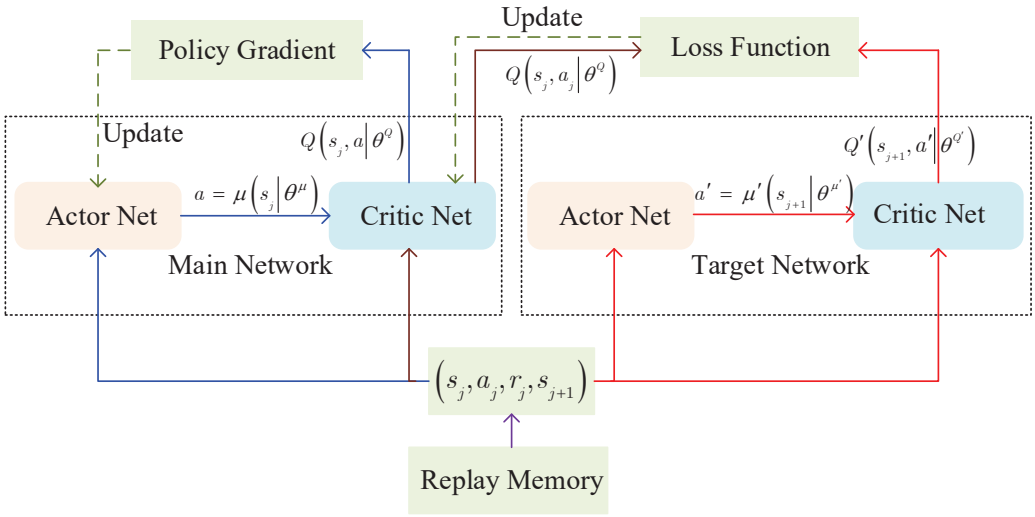

Figure 5. Framework of DIDDPG algorithm.

In each time slot j, the online actor network generates the corresponding action policy $\mu(s_j|\theta^\mu)$ based on state s_j. After executing the action $a_j = \mu(s_j|\theta^\mu) + \eta$ (η is an additional random noise to ensure the effective exploration), the next state s_{j+1} can be updated based on (11), and the corresponding reward r_j can be obtained according to (12). Then, the transition (s_j, a_j, s_{j+1}, r_j) is stored as a sample in the experience replay memory buffer. Repeating this process based on the closed loop control, enough training data can be generated by interacting with the environment. While training the networks, the mini-batch of K experience samples are randomly selected from the experience replay memory buffer in order to reduce the correlation among samples that the training efficiency can be improved.

By minimizing the loss function $L(\theta^Q)$, typically defined as a mean quadratic error function, the main critic network can update the parameter θ^Q using the gradient descent method:

$$L(\theta^Q) = \frac{1}{K}\sum_{j=0}^{K-1}\left(Q(s_j, a_j|\theta^Q) - y_j\right)^2, \tag{14}$$

where $Q(s_j, a_j|\theta^Q)$ represents the current Q value generated by the output of main critic network based on action a_j and state s_j, and y_j is the target Q value given by

$$y_j = r_j + \gamma Q'(s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'})|\theta^{Q'}). \tag{15}$$

In (15), $\mu'(s_{j+1}|\theta^{\mu'})$ and $Q'(s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'})|\theta^{Q'})$ denote the next action policy and next Q value derived from the target actor and critic networks, respectively.

Then, the main actor network updates the parameter θ^{μ} by the policy's gradient algorithm as [47]

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{K} \sum_{j=0}^{K-1} \left[\nabla_a Q(s, a|\theta^Q)|_{s=s_j, a=\mu(s_j)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_j} \right], \tag{16}$$

The updating gradient of the policy helps to improve the possibility of choosing a better action. Then, the DIDDPG softly updates the target networks as

$$\begin{aligned} \theta^{Q'} &\leftarrow \delta\theta^{Q} + (1-\delta)\theta^{Q'}, \\ \theta^{\mu'} &\leftarrow \delta\theta^{\mu} + (1-\delta)\theta^{\mu'}. \end{aligned} \tag{17}$$

and here, δ is a small constant.

After training, the parameters θ^{μ^*} will converge, and then the optimal formation control strategy for the follower is derived as

$$a^* = \mu(s|\theta^{\mu^*}). \tag{18}$$

The detailed DIDDPG-based UAV formation algorithm is presented as Algorithm 1.

Algorithm 1 DIDDPG-based UAV formation algorithm.

Initialize system parameters $\overline{P}, F_j, H_j, D_j^1, D_j^2$ and the replay memory buffer R.
Randomly initialize θ^{μ}, θ^{Q}, μ' and Q'.
Initialize online actor and critic networks $\mu(s|\theta^{\mu})$ and $Q(s, a|\theta^{Q})$, respectively.
for episode = $0:1:N-1$ **do**
  Initialize the random noise ω and state s_0.
  for $j = 0:1:M-1$ **do**
    Update the action $a_j = \mu(s_j|\theta^{\mu}) + \omega$.
    Update the next state s_{j+1} based on (11) that $s_{j+1} = s_j F_j + a_j H_j + \left[G_j^T, 0_{3\times 1}\right]$.
    Derive the reward r_j by (12) that $r_j = -s_j \bar{P} s_j^T - a_{j+1} Q a_{j+1}^T$.
    Store transition (s_j, a_j, r_j, s_{j+1}) in R.
    Randomly Select a mini-batch of K experience samples (s_j, a_j, r_j, s_{j+1}) from R.
    Update target Q value based on (15) that $y_j = r_j + \gamma Q'(s_{j+1}, \mu'(s_{j+1}|\theta^{\mu'})|\theta^{Q'})$.
    Update θ^{Q} by minimizing the mean quadratic error function based on (14).
    Update θ^{μ} by sampled policy gradient $\nabla_{\theta^{\mu}} J$ given by (16).
    Update the target networks:
    $\theta^{Q'} \leftarrow \delta\theta^{Q} + (1-\delta)\theta^{Q'}$,
    $\theta^{\mu'} \leftarrow \delta\theta^{\mu} + (1-\delta)\theta^{\mu'}$.
  end for
end for

3.3. Algorithm Analysis

The analysis of some issues, such as time delay and the computational complexity, are discussed for the proposed DIDDPG algorithm in this section.

3.3.1. Time Delay Analysis

Due to the inherent features of wireless transmission, the time delay is an inevitable issue that needs to be addressed in the UAV formation control process. It is known from (6) that the follower's state update is dependent on previous delayed control strategies due to the time delay. That is, the actor input in a sampling interval is given by

$$c(t) = \begin{cases} c_{j-1}, & j\Delta T < t \leq j\Delta T + \tau, \\ c_j, & j\Delta T + \tau < t \leq (j+1)\Delta T. \end{cases} \tag{19}$$

The different scenarios of time delay on the actor input are shown as in Figure 6. It is necessary to further discuss the influence of delayed information on how to design the DIMDP. Below, two special cases are represented to show the effect of time delay on the actor input, state definition and state transition function design of DIMDP.

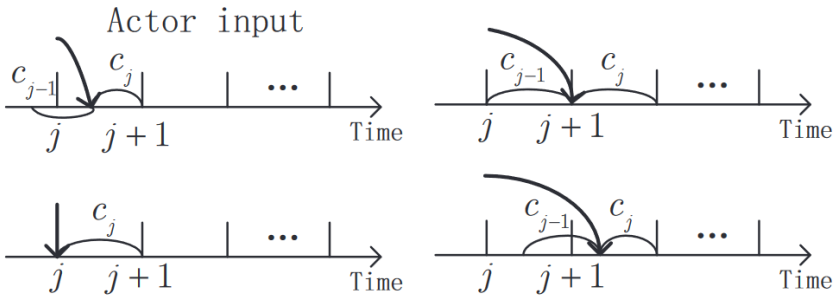

Figure 6. Different scenarios of actor input in a sampling interval.

When $\tau = 0$, the actor immediately receives the control strategy, and there is no effect of the previous control strategy on the follower's states. The discrete-time state update function is given by

$$z_{j+1} = Ez_j + D_j^1 c_j + G_j. \tag{20}$$

When $\tau = \Delta T$, the actor input only includes the previous control strategy in the j-th sampling interval, and the discrete state update function can be expressed as

$$z_{j+1} = Ez_j + D_j^2 c_{j-1} + G_j. \tag{21}$$

In fact, the time delay is influenced by many uncertainties, such as network topology, access technology and transmission channel quality, thus causing long and stochastic delays. Therefore, an arbitrary time delay should be further investigated, which is typically represented as $\tau \in [q\Delta T, (q+1)\Delta T)$, and here q is a positive integer [48]. Then, based on (5) and (6), the relevant discrete-time state update function can be expressed as

$$z_{j+1} = Ez_j + \tilde{D}_j^1 c_{j-q} + \tilde{D}_j^2 c_{j-q-1} + G_j, \tag{22}$$

where

$$D_j^1 = \int_0^{(q+1)T-\tau_j} e^{At} dt B, \; D_j^2 = \int_{(q+1)T-\tau_j}^{T} e^{At} dt B.$$

When the arbitrary time delay is considered, the follower's states are dependent on c_{j-q} and c_{j-q-1}. Similar to (8), the state can be extended to be

$$s_j = \left[{z_j}^T, {c_{j-1}}^T, {c_{j-2}}^T, \cdots, {c_{j-q-1}}^T \right]. \tag{23}$$

Then, based on (22) and (23), the state transition function can be formulated as

$$s_{j+1} = s_j \tilde{F}_j + a_j \tilde{H}_j + \left[G_j^T, 0_{3\times 1} \right], \tag{24}$$

where

$$\tilde{F}_j = \begin{bmatrix} E^T & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (\tilde{D}_j^1)^T & 0 & 0 & 0 & \cdots & 1 \\ (\tilde{D}_j^2)^T & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \tilde{H}_j = \begin{bmatrix} 0 \\ I_{3\times 3} \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T.$$

The reward function can be defined as

$$r_j = -\left(s_j \tilde{P} s_j^T + a_j Q a_j^T\right), \tag{25}$$

where

$$\tilde{P} = \begin{bmatrix} \overline{P} & 0_{6\times(q+1)} \\ 0_{(q+1)\times 6} & 0_{(q+1)\times(q+1)} \end{bmatrix}.$$

Based on the above extension definitions of state, the state transition function and reward function for arbitrary time delays, the proposed DIDDPG algorithm can be similarly applied to address the UAV formation control problem with long and stochastic delays.

3.3.2. Computational Complexity Analysis

In the following, the computational complexity, typically described as the floating point operations per second (FLOPS) of the training and validating processes for the proposed DIDDPG algorithm is investigated. In fact, the operation, such as multiplication and division, is regarded as a single FLOP. In the training process, the FLOPS can be derived as the computation times in actor and critic networks. In the validating process, only the main actor network needs to be considered because there is no replay buffer and critic network.

The computational complexity of the training process can be deduced as [49]

$$\begin{aligned} & v^{activation} u_i + 2 \times \sum_{m=0}^{M-1} u_m^{actor} u_{m+1}^{actor} + 2 \times \sum_{n=0}^{N-1} u_n^{critic} u_{n+1}^{critic} \\ & = O\left(\sum_{m=0}^{M-1} u_m^{actor} u_{m+1}^{actor} + \sum_{n=0}^{N-1} u_n^{critic} u_{n+1}^{critic}\right), \end{aligned} \tag{26}$$

where M and N are fully connected layers for the actor network and critic network, respectively. u_i means the unit number in the *i-th* layer, and $v^{activation}$ determined by the activation layer's type such that $v^{activation} = 1$, $v^{activation} = 4$ and $v^{activation} = 6$ represent the Relu layer, sigmoid layer and tanh layer, respectively.

During the validation process, only the main actor network exists. Then, the computational complexity for the validation process is given by

$$O\left(\sum_{n=0}^{N-1} u_n^{critic} u_{n+1}^{critic}\right). \tag{27}$$

In the proposed DIDDPG-based UAV formation algorithm, double fully connected layers with 30 units and 1 units, respectively, are used to build the actor network, and Relu and tanh layer are used as the activation layer. Double fully connected layers with 60 units and 1 units, respectively, are used to build the critic network, and the Relu layer is used as the activation layer. Based on (26) and (27), the computations of the actor network and critic network are obtained as 756 and 900, respectively.

4. Simulation Results and Discussions

Numerical experiments are presented in this section to evaluate the performance of DIDDPG algorithm. The flight data are designed based on real UAV flight data in [29,30].

First, we show the effectiveness and convergence of the proposed DIDDPG algorithm. Then, we compare proposed algorithm with existing algorithms for performance evaluation. Last, it is verified that the proposed optimal policies are applicable to long arbitrary time delays. As a case study, a typical 2D UAV formation with constant altitude is investigated.

In order to avoid collisions and improve the formation, the desired velocity and headway (i.e., the relative distance between the leader and the follower) are often influenced by each other. Typically, the expected headway needs to be adjusted in real time according to the UAV velocity change, that is, the expected headway will become larger with the increase in the desired UAV velocity. As an example for simulations, we set this relationship as a typical sigmoid function as [50]

$$v(h) = \begin{cases} 0, & 0 < h < h_{\min} \\ \frac{v_{\max}}{2}(1 - \cos(\pi \frac{h - h_{\min}}{h_{\max} - h_{\min}})), & h_{\min} \le h \le h_{\max} \\ v_{\max}, & h > h_{\max} \end{cases} \tag{28}$$

where h denotes the headway, h_{min} and h_{max} represent the maximum and minimum headway, respectively, and v_{max} means the maximum velocity.

In the simulations, the system parameter settings are presented as in Table 1.

Table 1. Simulation parameter settings.

Symbol	Description	Setting
S^v	Velocity space	$[0, 30]$ m/s
S^p	Acceleration space	$[-5, 5]$ m/s^2
K	Mini-batch size	32
N	Episode	260
M	Time steps	200
l_a, l_c	Learning rates for actor and critic	0.001, 0.002
γ	Discount factor	0.97
h_{max}	Maximum headway	30
h_{min}	Minimum headway	5
ΔT	Sampling interval	0.2 s

4.1. Performance Comparison of Convergence

The convergence of the proposed DIDDPG algorithm is evaluated and analyzed under various reward function forms and learning rates, and time delay is uniform in $[0, 0.2\Delta T]$. In order to facilitate performance comparison, we take the following normalization measure to the cumulative rewards as

$$\bar{G}_N = \frac{G_N - G_{\min}}{G_{\max} - G_{\min}}, \tag{29}$$

where $G_{\max}$ and $G_{\min}$ are the maximum and minimum cumulative rewards, respectively.

Figures 7 and 8 depicts the convergence of the proposed intelligent control algorithm under different actor and critic learning rates when the reward function is quadratic. If the learning rate is too small, the gradient descent could be slow, or the gradient descent may overshoot the minimum value such that it will fail to converge or even diverge. Obviously, in the case $l_c = 0.00002$, the parameter update speed is slow, resulting in the inability to quickly find a good descending direction. Thus, the suitable range of values of l_a and l_c when the reward function is quadratic is obtained. In Figure 9, the effects of three types of reward functions under suitable learning rate values from Figures 7 and 8 on the convergence performance of proposed algorithm are compared. It can be observed that the learning process of the case of quadratic reward function is the fastest and most stable. It indicates that, within suitable learning rates, the quadratic reward function consistently

outperforms other forms and achieves the most benefit for the purposed intelligent control algorithm because it is consistent with the cost function of the UAV formation.

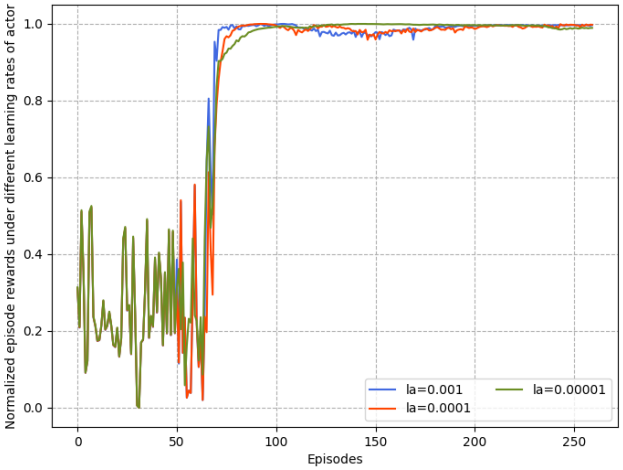

Figure 7. Normalized reward comparison with different learning rates of actor.

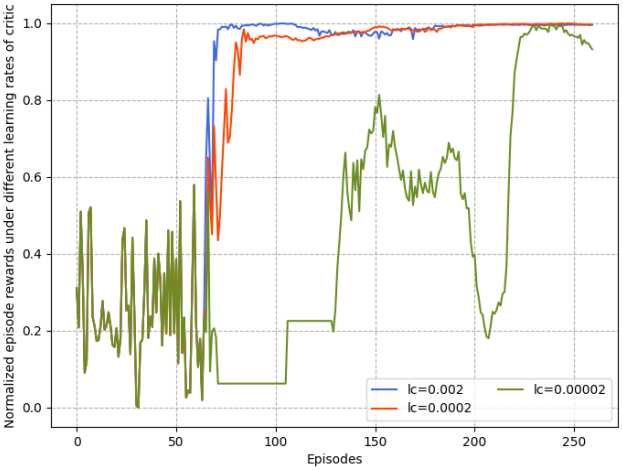

Figure 8. Normalized reward comparison with different learning rates of critic.

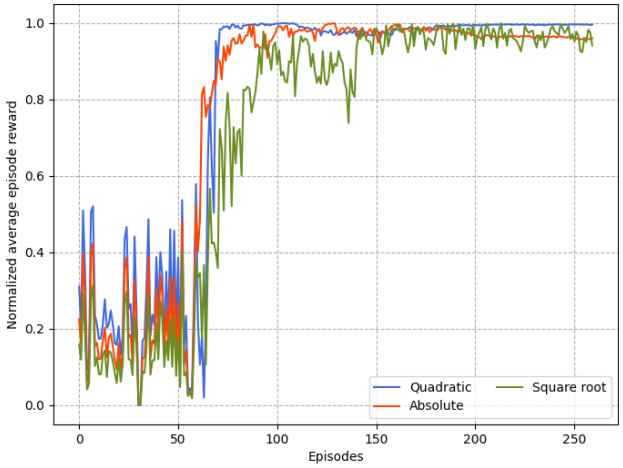

Figure 9. Normalized reward comparison with different forms of reward function.

4.2. Performance Comparison of Different Scenarios

The velocity and headway tracking performance in the presence of different time delays under different application scenarios are shown in Figures 10–12. In the simulations,

three scenarios are considered, including harsh brake, stop-and-go and speed limit. These three basic cases are covered by most application tasks, and the proposed algorithm has good practicality if it can satisfy the control requirements in these three cases. The simulation results show that the follower can track the desired states accurately by the proposed algorithm. Figure 10 shows the case when the follower suddenly meets an obstacle and needs to brake harshly, and the rapid velocity decline happens to represent the harsh brake. It takes about 11 s for the follower to stop from 20 m/s. Figure 11 shows the application scenario when UAV needs to stop and hover sometimes. For example, the UAV-assisted wireless powered IoT network, where UAVs hover to visit IoT devices and collect data, and the velocity variations are typically small. It can be seen that near 10 s, the velocity reaches the desired value and the headway stops changing; although there is a small error between the desired states, it is still within the acceptable range, and at 14 s when the follower starts flying, it can quickly follow the desired state. Figure 12 shows the case that UAV flights in restricted environments and the velocity change are limited. What is more, the headway's tendency over time is the same as that of velocity, which is consistent with the relationship of the headway and velocity. The results show that proposed intelligent algorithm could be applicable to either high- or low-velocity cases and also could be used in large and small velocity variation conditions. In general, the proposed algorithm can derive the control strategy satisfying the tracking assignment under the above three common scenarios.

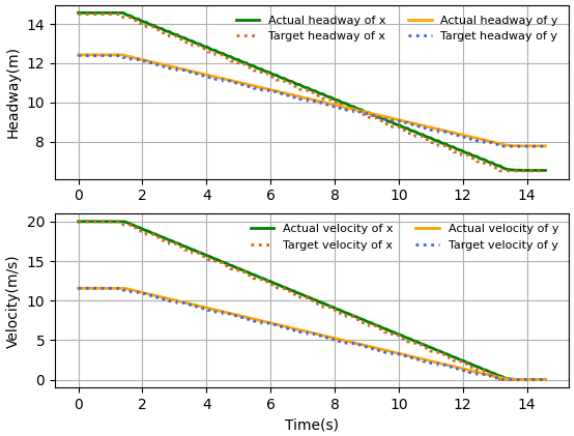

Figure 10. Harsh brake.

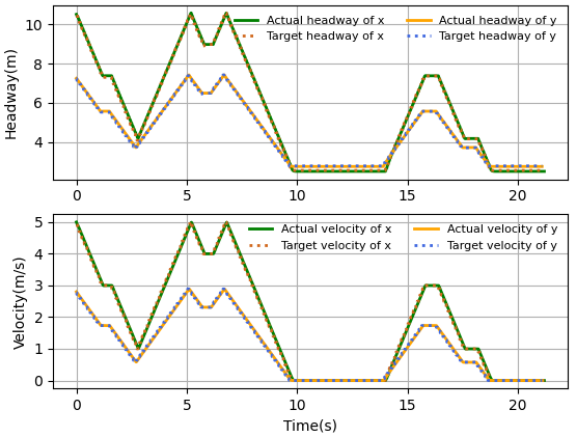

Figure 11. Stop-and-go.

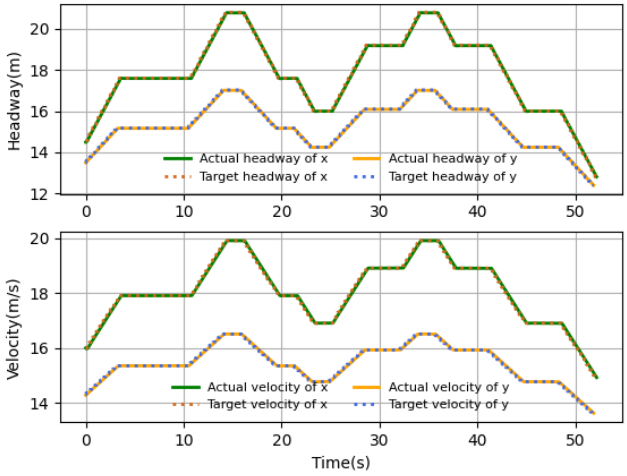

Figure 12. Speed limit.

4.3. Performance Comparison with Different Aspects

The performance comparisons with different time delays and existing algorithms are shown in Figures 13 and 14, respectively.

In Figure 13, the time delay is set to be $0.2\Delta T$, ΔT and $1.5\Delta T$, and here the deterministic time delay settings represent the three delay scenarios discussed in Section 3.3.1 to demonstrate the influence of time delays on the relative performance of the proposed algorithm. Figure 13 shows that a larger time delay leads to more serious performance degradation. For example, when the time delay $\tau = 1.5\Delta T$, the control strategy executed in each sampling interval is the delayed control strategy but not the current control strategy, thus causing the followers to react slower. Fortunately, the control performance still meets the tracking requirement. It indicates that the proposed algorithm can effectively regulate the follower to achieve the stable tracking under various time delays. When the time delay $\tau = 0.2\Delta T$, the follower can keep close to the desired states all the time, which indicates that the proposed algorithm can compensate for the effect of the time delay and improve the control performance.

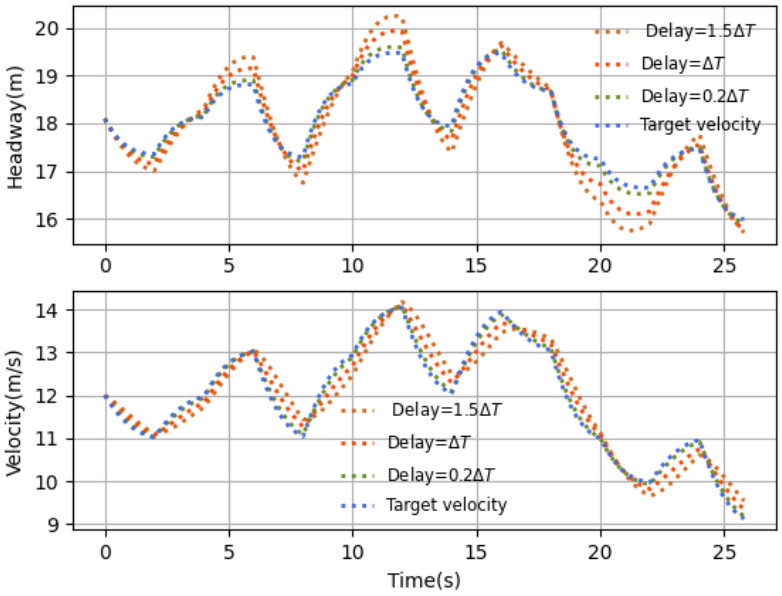

Figure 13. Velocity comparison with different delays.

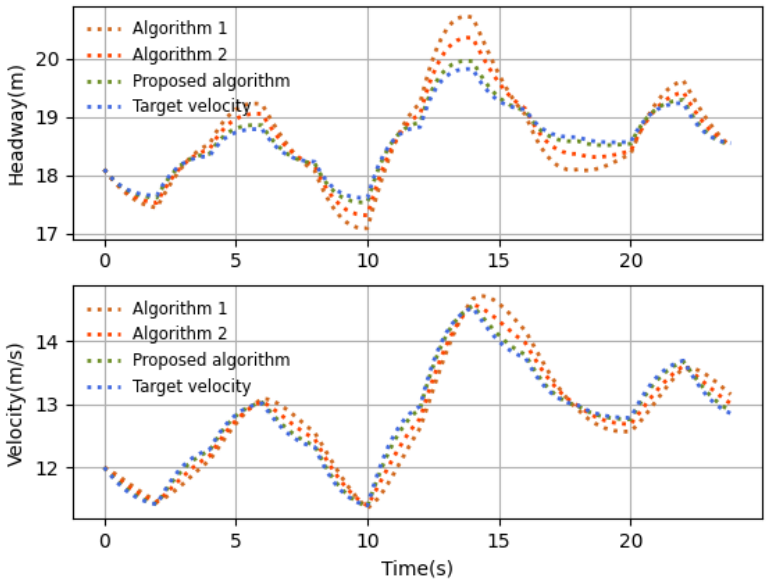

Figure 14. Velocity comparison with existing algorithms (Algorithm 1 in [39] and Algorithm 2 in [43]).

Figure 14 shows that our proposed algorithm has the quickest response and best control performance compared with the existing works under the time-varying leader velocity. In the simulations, the sampling interval is $\Delta T = 0.2$ s, the time delay is uniform in $[0, 0.2\Delta T]$, and the other system parameter settings are the same as those in Table 1. Actually, the existing algorithm in [39] does not include the previous actions into the state, which may lead to the insufficient utilization of the delay information. Therefore, although this existing algorithm can reach the desired states, it still reacts more slowly. The existing algorithm in [43] does not consider the latency information in the agent environment, resulting in its performance being worse than the others.

5. Conclusions

UAV formation can be deployed in a multitude of surveillance scenarios. The leader–follower approach can effectively improve the efficiency of the whole formation. Since the desired velocity and time delay are dynamic due to different scenarios and the inherent feature of wireless communications, it is taken into account in the optimization formation problem in this paper. In order to compensate for the effect of time delay, a new MDP, called DIMDP, is designed by including previous actions into the state and reward function, and then the DIDDPG algorithm is proposed to solve the DIMDP of the UAV formation. The reward function form is designed, dependent on the quadratic cost function relevant to the objective of the optimization formation problem. After training, the intelligent control strategy can be derived for the follower. The simulation experiments demonstrate that the proposed intelligent controller can effectively alleviate the effects of time delays and is applicable to high dynamic formation scenarios. Compared with existing DRL algorithms with or without time delays, the proposed DIDDPG algorithm can achieve better control convergence and stability. However, the proposed algorithm is designed based on the flight data in the simulation according to the existing literature, and the lack of the real-world data or realistic simulation environments needs to be addressed in future work. The cooperative formation control system considered and designed in this paper aims to achieve control of the entire formation by dividing it into individual units and realizing the tracking control of each LF unit. However, at present, the construction of the multi-UAV cooperative control system and the research on multi-objective control algorithms are gradually attracting attention, and the multi-intelligent reinforcement learning algorithm can be studied in the future.

Author Contributions: Conceptualization, L.L., M.X., Z.W. and C.F.; methodology, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; software, L.L., M.X., Z.W. and C.F.; validation, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; formal analysis, L.L., M.X., Z.W. and C.F.; investigation, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; data curation, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; writing—original draft preparation, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; writing—review and editing, L.L., M.X., Z.W., C.F., Z.L., M.L., Y.S. and H.C.; project administration, L.L. and Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Beijing Natural Science Foundation 4222002, L202016, and L211002, in part by the Beijing Nova Program of Science and Technology Z191100001119094, in part by the Foundation of Beijing Municipal Commission of Education KM202110005021, and in part by the Urban Carbon Neutral Science and Technology Innovation Fund Project of Beijing University of Technology 040000514122607.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

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

References

1. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [CrossRef]
2. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
3. Cho, J.; Sung, J.; Yoon, J.; Lee, H. Towards persistent surveillance and reconnaissance using a connected swarm of multiple UAVs. *IEEE Access* **2020**, *8*, 157906–157917. [CrossRef]
4. Jasim, W.; Gu, D. Robust team formation control for quadrotors. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 1516–1523. [CrossRef]
5. Chao, Z.; Zhou, S.L.; Ming, L.; Zhang, W.G. UAV formation flight based on nonlinear model predictive control. *Math. Probl. Eng.* **2012**, *2012*, 261367. [CrossRef]
6. Chao, Z.; Ming, L.; Shaolei, Z.; Wenguang, Z. Collision-free UAV formation flight control based on nonlinear MPC. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 1951–1956.
7. Cordeiro, T.F.K.; Ferreira, H.C.; Ishihara, J.Y. Non linear controller and path planner algorithm for an autonomous variable shape formation flight. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1493–1502.
8. Najm, A.A.; Ibraheem, I.K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1087–1097. [CrossRef]
9. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
10. Jiang, Y.; Jiang, Z.P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* **2012**, *48*, 2699–2704. [CrossRef]
11. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
12. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [CrossRef]
13. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
14. Li, B.; Wu, Y. Path planning for UAV ground target tracking via deep reinforcement learning. *IEEE Access* **2020**, *8*, 29064–29074. [CrossRef]
15. Bayerlein, H.; Theile, M.; Caccamo, M.; Gesbert, D. UAV path planning for wireless data harvesting: A deep reinforcement learning approach. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
16. Koch, W.; Mancuso, R.; West, R.; Bestavros, A. Reinforcement learning for UAV attitude control. *ACM Trans.-Cyber-Phys. Syst.* **2019**, *3*, 1–21. [CrossRef]
17. Buechel, M.; Knoll, A. Deep reinforcement learning for predictive longitudinal control of automated vehicles. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2391–2397.

18. Chu, T.; Kalabi, U. Model-based deep reinforcement learning for CACC in mixed-autonomy vehicle platoon. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 4079–4084.
19. Bouhamed, O.; Ghazzai, H.; Besbes, H.; Massoud, Y. Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach. In Proceedings of the 2020 IEEE International Symposium on circuits and systems (ISCAS), Virtual Event, 10–21 October 2020; pp. 1–5.
20. Wan, K.; Gao, X.; Hu, Z.; Wu, G. Robust motion control for UAV in dynamic uncertain environments using deep reinforcement learning. *Remote Sens.* **2020**, *12*, 640. [CrossRef]
21. Wen, G.; Chen, C.L.P.; Feng, J.; Zhou, N. Optimized multi-agent formation control based on an identifier-actor-critic reinforcement learning algorithm. *IEEE Trans. Fuzzy Syst.* **2017**, *26*, 2719–2731. [CrossRef]
22. Yang, Y.; Modares, H.; Wunsch, D.C.; Yin, Y. Leader-follower output synchronization of linear heterogeneous systems with active leader using reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2139–2153. [CrossRef] [PubMed]
23. Zhao, Y.; Ma, Y.; Hu, S. USV formation and path-following control via deep reinforcement learning with random braking. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5468–5478. [CrossRef]
24. Liu, Y.; Bucknall, R. A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica* **2018**, *36*, 1019–1047. [CrossRef]
25. Wang, X.; Yadav, V.; Balakrishnan, S.N. Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 672–679. [CrossRef]
26. Wang, J.; Xin, M. Integrated optimal formation control of multiple unmanned aerial vehicles. *IEEE Trans. Control Syst. Technol.* **2012**, *21*, 1731–1744. [CrossRef]
27. Kuriki, Y.; Namerikawa, T. Consensus-based cooperative formation control with collision avoidance for a multi-UAV system. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 2077–2082.
28. Dong, X.; Yu, B.; Shi, Z.; Zhong, Y. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 340–348. [CrossRef]
29. Dong, X.; Zhou, Y.; Ren, Z.; Zhong, Y. Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5014–5024. [CrossRef]
30. Yan, C.; Xiang, X.; Wang, C. Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. *J. Intell. Robot. Syst.* **2020**, *98*, 297–309. [CrossRef]
31. Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J.J.; Gupta, A.; Fei-Fei, L.; Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3357–3364.
32. Walsh, T.J.; Nouri, A.; Li, L.; Littman, M.L. Learning and planning in environments with delayed feedback. *Auton. Agents Multi-Agent Syst.* **2009**, *18*, 83–105. [CrossRef]
33. Adlakha, S.; Madan, R.; Lall, S.; Goldsmith, A. Optimal control of distributed Markov decision processes with network delays. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 3308–3314.
34. Zhong, A.; Li, Z.; Wu, D.; Tang, T.; Wang, R. Stochastic peak age of information guarantee for cooperative sensing in internet of everything. *IEEE Internet Things J.* **2023**, 1–10. [CrossRef]
35. Ramstedt, S.; Pal, C. Real-time reinforcement learning. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 1–10.
36. Li, Z.; Li, F.; Tang, T.; Zhang, H.; Yang, J. Video caching and scheduling with edge cooperation. *Digit. Commun. Netw.* **2022**, 1–13. [CrossRef]
37. Zhao, H.; Wang, B.; Liu, H.; Sun, H.; Pan, Z.; Guo, Q. Exploiting the flexibility inside park-level commercial buildings considering heat transfer time delay: A memory-augmented deep reinforcement learning approach. *IEEE Trans. Sustain. Energy* **2021**, *13*, 207–219. [CrossRef]
38. Nath, S.; Baranwal, M.; Khadilkar, H. Revisiting state augmentation methods for reinforcement learning with stochastic delays. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management, Virtual Event, 1–5 November 2021; pp. 1346–1355.
39. Chen, B.; Xu, M.; Li, L.; Zhao, D. Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing* **2021**, *450*, 119–128. [CrossRef]
40. Li, Z.; Zhu, N.; Wu, D.; Wang, H.; Wang, R. Energy-efficient mobile edge computing under delay constraints. *IEEE Trans. Green Commun. Netw.* **2021**, *6*, 776–786. [CrossRef]
41. Zeng, T.; Semiari, O.; Saad, W.; Bennis, M. Joint communication and control for wireless autonomous vehicular platoon systems. *IEEE Trans. Commun.* **2019**, *67*, 7907–7922. [CrossRef]
42. Li, Z.; Zhou, Y.; Wu, D.; Tang, T.; Wang, R. Fairness-aware federated learning with unreliable links in resource-constrained Internet of things. *IEEE Internet Things J.* **2022**, *9*, 17359–17371. [CrossRef]
43. Wang, C.; Wang, J.; Shen, Y.; Zhang, X. Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2124–2136. [CrossRef]
44. Zhang, A.; Zhou, D.; Yang, M.; Yang, P. Finite-time formation control for unmanned aerial vehicle swarm system with time-delay and input saturation. *IEEE Access* **2018**, *7*, 5853–5864. [CrossRef]

45. Gonzalez, A.; Aranda, M.; Lopez-Nicolas, G.; Sagüés, C. Time delay compensation based on Smith predictor in multiagent formation control. *IFAC-PapersOnLine* **2017**, *50*, 11645–11651. [CrossRef]
46. Su, H.; Wang, X.; Lin, Z. Flocking of multi-agents with a virtual leader part II: With a virtual leader of varying velocity. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 1429–1434.
47. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Detroit, MI, USA, 3–5 December 2014; pp. 387–395.
48. Wang, Z.; Gao, Y.; Fang, C.; Liu, L.; Guo, S.; Li, P. Optimal connected cruise control with arbitrary communication delays. *IEEE Syst. J.* **2019**, *14*, 2913–2924. [CrossRef]
49. Qiu, C.; Hu, Y.; Chen, Y.; Zeng, B. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. *IEEE Internet Things J.* **2019**, *6*, 8577–8588. [CrossRef]
50. Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805. [CrossRef] [PubMed]