aerospace

MDPI

Article

Autonomous Shape Decision Making of Morphing Aircraft with Improved Reinforcement Learning

Weilai Jiang [1,2], Chenghong Zheng [1,2,*], Delong Hou [3], Kangsheng Wu [4] and Yaonan Wang [1,2]

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China; jiangweilai@hnu.edu.cn (W.J.); yaonan@hnu.edu.cn (Y.W.)
[2] Greater Bay Area Institute for Innovation, Hunan University, Guangzhou 511300, China
[3] Beijing Electronic Engering System Insitute, Beijing 100854, China; houdelong@casic.com.cn
[4] Zoomlion Heavy Industry Science and Technology Co., Ltd., Changsha 410082, China; 17877783998@163.com
* Correspondence: zhengchenghong@hnu.edu.cn

Abstract: The autonomous shape decision-making problem of a morphing aircraft (MA) with a variable wingspan and sweep angle is studied in this paper. Considering the continuity of state space and action space, a more practical autonomous decision-making algorithm framework of MA is designed based on the deep deterministic policy gradient (DDPG) algorithm. Furthermore, the DDPG with a task classifier (DDPGwTC) algorithm is proposed in combination with the long short-term memory (LSTM) network to improve the convergence speed of the algorithm. The simulation results show that the shape decision-making algorithm based on the DDPGwTC enables MA to adopt the optimal morphing strategy in different task environments with higher autonomy and environmental adaptability, which verifies the effectiveness of the proposed algorithm.

Keywords: morphing aircraft; reinforcement learning; deep deterministic policy gradient; long short-term memory network; shape decision making

Citation: Jiang , W.; Zheng, C.; Wu, K.; Wang, Y. Autonomous Shape Decision Making of Morphing Aircraft with Improved Reinforcement Learning. *Aerospace* **2024**, *11*, 74. https://doi.org/10.3390/aerospace11010074

Academic Editor: Rosario Pecora

Received: 22 October 2023
Revised: 14 December 2023
Accepted: 9 January 2024
Published: 12 January 2024

1. Introduction

In recent years, both military and civil fields have put forward higher requirements for the stsability, autonomy and reliability of the next generation of aircraft, which should have the ability to maintain stability under different flight environments and tasks. Traditional fixed-wing aircraft are only designed for specific flight conditions, which cannot meet the requirements of various tasks in changing environments, while morphing aircraft (MA) can achieve optimal flight performance by changing its shape structure to adapt to different environments and tasks [1]. The adaptability of MA can be effectively improved by appropriate autonomous morphing strategy, which has been widely studied by scholars at home and abroad [2–5].

In the control discipline, the research of MA can be divided into three main levels: morphing control, flight control and shape decision making. Among them, the research into morphing control mostly focuses on the morphing airfoil structures, flight dynamics and intelligent materials [6–9]. The research into flight control assumes that the aircraft shape changes according to the preset change strategy [10–13]. Obviously, the assumption that the shape changes passively does not meet people's pursuit of the "intelligent and adaptive" property of MA. The research into shape decision making is a difficult problem in the field of MA control because of the complex tasks and environments. Therefore, there are few related research results. The emergence of artificial intelligence provides a new idea for the research of autonomous shape decision-making. Recently, the deep reinforcement learning (DRL) policy proposed by Google Deepmind can perceive environmental change, interact with the environment via trial and error mechanism, and learn the best policy by maximizing the cumulative reward. The results were verified on AlphaGo Zero [14–18]. In order to realize shape decision making of MA in the actual flight environment, John Valasek of Texas A&M

University first introduced reinforcement learning (RL) algorithms of artificial intelligence into morphing control [19,20]. In [19], he replaced the MA model with a three-dimensional cuboid and used the actor–critic (AC) algorithm in RL to solve the optimal shape of the aircraft. However, due to the limitations of the AC algorithm, the convergence of the results is difficult. Then, in [20], he used the continuous function approximation method instead of the K-nearest neighbors algorithm (KNN) method, and adopted the Q-learning algorithm combined with adaptive dynamic inverse control to obtain a better control effect. Lampton et al. [21–23] considered a type of National Advisory Committee for Aeronautics (NACA) deformable wings, and applied unsupervised learning to Q-learning to obtain the optimal shape of wings with different lift, drag and torque coefficient requirements. In [24], a new airfoil adaptive control method for MA is proposed by using a Q-learning algorithm and sliding mode control. The Q-learning algorithm in [20–24] can only be used in the discrete state and action space, but the process of changing the shape of the aircraft is a continuous action, so it is difficult to apply to engineering practice. The deep deterministic policy gradient (DDPG) algorithm can be used in the continuous state and action space [25], which is a better solution to the shape decision-making problem of MA. In [26], an abstract MA is used as the object, and its shape change equation and optimal shape function are given. But this method only considers the ellipsoid as MA and does not consider the true MA, so it only has theoretical significance. In [27,28], the semi-physical simulation experiment of the morphing wing is carried out based on the DDPG algorithm, and the MA model is controlled to complete the required morphing tasks. But the research adopts a simplified aircraft model and only uses the semi-physical simulation for theoretical verification without combining the actual flight conditions. At present, most of the existing autonomous shape decision-making strategies of MA use the simplified MA model or the RL algorithm of discrete action space. There is no research on applying the RL method of continuous action space to a real MA model, which leads to some limitations in the application of the existing results to engineering practice.

To overcome the shortcomings of the existing research results, this paper studies the autonomous shape decision-making problem of MA with an improved RL algorithm. The main works and contributions are as follows. Firstly, it is different from the Q-learning algorithm used in [20–24], which can only work on discrete state and action space. In this paper, considering the continuity of state space and action space of the aircraft, the DDPG algorithm is used to establish a more practical shape decision-making algorithm framework for MA. Furthermore, in order to improve the convergence of the algorithm, a task classifier is designed combined with a long short-term memory (LSTM) network. The DDPG with a task classifier (DDPGwTC) algorithm is proposed as the shape optimization strategy of MA, which improves the convergence speed of the algorithm. Finally, the simulation results show that the proposed algorithm can make the shape decision making of MA autonomously.

The following is the arrangement of this article. Section 2 focuses on shape decision making of MA based on the DDPGwTC algorithm. In Section 3, and the network training of task classifier and DDPGwTC algorithm is discussed. Section 4 and Section 5 give the simulation analysis and the conclusion of this paper, respectively.

2. Shape Decision Making of MA Based on the DDPGwTC Algorithm

2.1. Principles of the DDPG Algorithm

MA is in a complex and changeable flight environment. The state of MA will change in real time, which requires its shape to change rapidly and continuously. Q-learning [29] is a typical RL algorithm based on the value function, which is only suitable for discrete state and action space. The traditional deep Q network (DQN) algorithm based on Q-learning can only solve the problem of discrete action space, and the DPG (deterministic policy gradient) [18] algorithm is also faced with the problem of difficult exploration. Therefore, they are not suitable for the shape decision making of MA. DDPG [25] algorithm is based on the DPG algorithm, which adopts the framework of the AC algorithm and inherits the

advantages of the DQN algorithm. It can effectively solve the RL problem in continuous action space. The DDPG algorithm uses a deep neural network (DNN) to approximate the policy function and value function, and obtains the optimal policy by maximizing the reward function. Then, the policy network generates the actual action. The update approach is as follows [25].

$$\nabla_{\theta^\mu} J \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_a Q(s,a|\theta^Q)\Big|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)\Big|_{s_t} \tag{1}$$

The value network evaluates the action of the policy network by fitting the action value function and its updating approach is the same as the value-based function. The network parameter is updated by minimizing the loss function and the equation is as follows.

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{Q'}\right) - Q\left(s_i, a_i\middle|\theta^{Q}\right)\right)^2 \tag{2}$$

Using the DDPG agent as the shape decision-making controller of MA and giving the corresponding morphing strategy based on the output data of the aircraft is the general idea of the research. However, there are still many difficulties in applying the DDPG algorithm to MA. How to establish the DDPG algorithm model so it is applicable to MA is a very critical step, which includes the definition of the environment model, the state space and the action space, and the design of the reward function.

2.2. Design of the DDPG Algorithm with Task Classifier

2.2.1. Framework of Shape Decision-Making Algorithm for MA

This article studies a variable sweep angle and wingspan aircraft, which is based on the Navion-L17 aircraft. As shown in Figure 1, the wing span deformation rate is defined as $\lambda = (l - l_{\min})/(l_{\max} - l_{\min})$, where l is the actual span, l_{min} and l_{max} are the shortest and longest spans, respectively. The longest span can reach twice the shortest span, which is easily known as $\lambda \in [0,1]$. The defined range of wing sweep angle variation is between 0 and 40 °, with a sweep angle deformation rate of $\rho = \eta/40$, where ρ is the actual sweep angle, which is easily known as $\rho \in [0,1]$. The wingspan and sweep angle can be set to continuously change simultaneously.

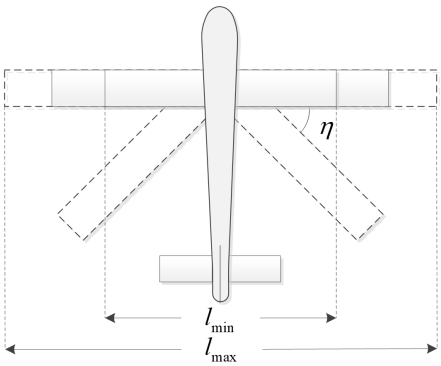

Figure 1. The shape decision-making algorithm framework of MA.

During the deformation process, changes in wingspan and sweep angle can cause changes in parameters such as the aircraft's center of gravity and moment of inertia, resulting in changes in the aerodynamic forces and moments acting on it. Therefore, this article uses a multi rigid body modeling method to treat the variant aircraft as a multi rigid body system composed of deformable wings and fuselage, and to model the dynamics of the aircraft. The modeling rules follow the following assumptions:

(1) The aircraft fuselage is symmetrical about the longitudinal plane of the aircraft coordinate system, and the process of wing modification on both sides is synchronized. During the modification process, the center of mass of the aircraft moves along the x-axis of the aircraft, only considering the influence on longitudinal motion and not producing a component that affects lateral motion.
(2) The aircraft adopts a single engine, ignoring the influence of the component generated by the engine installation angle on thrust and the influence of thrust on pitch torque.
(3) Set the gravitational acceleration as a constant, ignoring changes in the mass of the aircraft.
(4) Neglect the impact of unsteady aerodynamic forces generated during the variant process on the aircraft.

By combining the kinematic equations of traditional aircraft, a nonlinear model of longitudinal motion of the variant aircraft can be obtained:

$$\begin{cases} \dot{V} = \frac{T}{m}\cos\alpha - \frac{1}{m}D(\lambda,\rho) - g\sin(\theta-\alpha) \\ \dot{\alpha} = -\frac{T}{mV}\sin\alpha - \frac{1}{mV}L(\lambda,\rho) + q + \frac{g}{V}\cos(\theta-\alpha) \\ \dot{\theta} = q \\ \dot{q} = \frac{1}{I_y}M(\lambda,\rho) \\ \dot{h} = V\sin(\theta-\alpha) \end{cases} \tag{3}$$

where h is the flight altitude. Thrust T, lift L, drag D, and pitch moment M can be further expressed as:

$$\begin{cases} L = \frac{1}{2}\rho' V^2 S_W C_L(\lambda,\rho) \\ D = \frac{1}{2}\rho' V^2 S_W C_D(\lambda,\rho) \\ M = \frac{1}{2}\rho' V^2 S_W c_A C_m(\lambda,\rho) \\ T = T_{\delta_t}\delta_t \end{cases} \tag{4}$$

where ρ' represents the atmospheric density at a certain altitude. Since the altitude does not change much during the simulation process, this article considers it as a constant, where S_w represents the wing area, which varies with random wing deformation C_L, C_D, C_m represents lift coefficient, drag coefficient, and pitch moment coefficient, respectively, C_A is the average aerodynamic chord length, $T_{\delta_t} = 50N/\%$ is the thrust coefficient, and δ_t is the throttle opening.

It should be noted that the control of MA includes the outer loop shape decision and the inner loop attitude control. This paper considers a class of MA with variable wingspan and variable sweep angle, and focuses on the shape decision making of MA. However, the attitude control of MA is the basic condition of shape decision-making. Therefore, the conclusion of flight control of MA based on the T-S fuzzy model is given. Readers can refer to [30,31] for more details. The T-S fuzzy model is as follows.

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{6} \mu_i(\lambda) A_i x(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{5}$$

where $x(t)$ is the state variables of MA, including speed V, altitude h, angle of attack α, angle of pitch θ and pitch angular velocity q, $u(t)$ is the control variables of MA, including angle of elevator deflection $\delta_e$and throttle opening δ_t, $y(t)$ is the output, A, B, C, D are system matrixes and $\mu_i(\lambda)$ is the activation degree of fuzzy rules.

To enable MA to track the preset command and keep stable during the morphing process, it is necessary to design a suitable tracking controller, which mainly achieves two goals. One is to require the closed-loop system of the aircraft to keep stable without steady-state error at any λ. The other is to keep the closed-loop system stable in the morphing process. For the T-S fuzzy model in Equation (5), the parallel distributed compensation

(PDC) fuzzy controller design approach is adopted. The basic principle is to design a linear controller for each local linear model then connect the linear controller with the same fuzzy rules as the local linear model to obtain the global T-S fuzzy controller, as shown in Equation (6).

$$u = \sum_{i=1}^{6} \mu_i(\lambda) K_i x(t) \tag{6}$$

where K_i is the designed local linear controller.

Substituting Equation (6) into Equation (5), the augmented closed-loop system with T-S fuzzy controller *can be obtained as follows.*

$$\begin{cases} \dot{x}(t) = \sum\limits_{i=1}^{6} \sum\limits_{j=1}^{6} \mu_i(\lambda)\mu_j(\lambda)\left(A_i + BK_j\right)x(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{7}$$

Combining the above model with the DDPG algorithm, the shape decision-making algorithm framework of MA based on the DDPG can be obtained as shown in Figure 2.

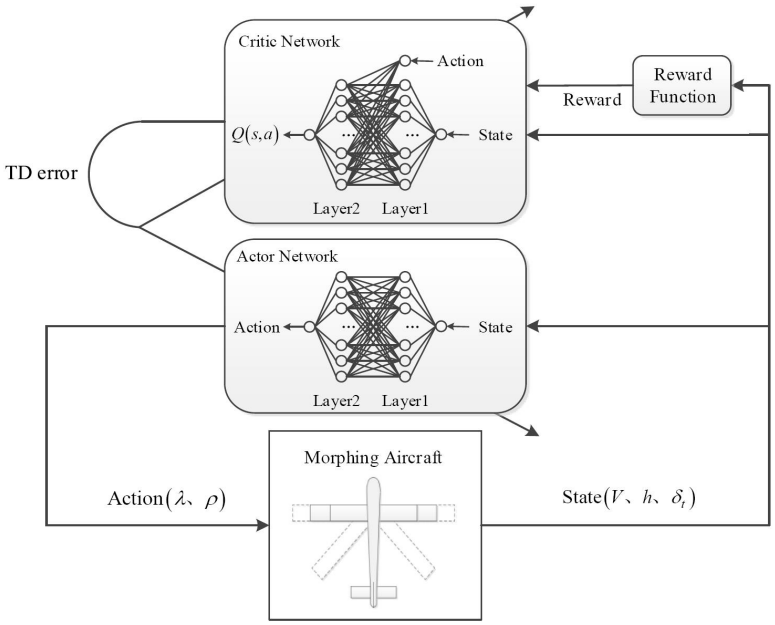

Figure 2. The shape decision-making algorithm framework of MA.

As shown in Figure 2, the environment model of the algorithm is the MA control system shown in Equation (7). The aircraft changes its wingspan and sweep angle according to the morphing strategy obtained by the agent, and the state of MA is fed back to the agent. The wingspan variation rate λ and sweep angle variation rate ρ of MA are defined as the action space. The altitude, speed and throttle opening of MA are defined as the state space, which can be obtained by sensors. Obviously, the action and state space are continuously changing values. The design of the reward function is the key to whether the RL algorithm can converge to the optimal policy, because the agent adjusts the action according to the reward feedback from the environment model. The optimal shape of MA should be to minimize the tracking error and fuel consumption of altitude and speed, so the reward function can be designed by using the time error integral index.

There are two problems in applying the DDPG algorithm to the autonomous shape decision making of MA. First, MA is in a complex flight environment, and the change of wingspan or sweep angle will have a certain impact on the flight state. The external gust interference will also affect the tilt angle of the aircraft, resulting in the instability of speed and altitude. Even though this instability is greatly reduced with the controller,

but the task state of the aircraft cannot be accurately determined and the reward function of the algorithm cannot be accurately set. Therefore, it may take a long time for the DDPG algorithm to learn a stable policy; however, the final policy is not necessarily optimal. Moreover, the random selection of action in the training process will lead to a large oscillation of the aircraft system, resulting in the low learning efficiency of the algorithm in the initial stage. Finally, it is difficult to converge or converge to the suboptimal policy.

To solve the above problems, this paper proposes a DDPGwTC algorithm based on the DDPG algorithm which combines the advantages of LSTM in dealing with sequence classification problems. Its pseudocode can be found in Algorithm 1. This algorithm adds a task classifier before the agent obtains the feedback state of the aircraft, classifies the state of the aircraft into different task types through LSTM, and inputs the task type as the state of the algorithm into the agent to guide the agent to make the optimal action. The reward function is also set into different forms according to different task types. This improvement approach is equivalent to adding some "prior knowledge" to the agent, which enables the agent to classify the regular data in the unstable state, reduces the trial and error cost of the algorithm and improves the convergence speed of the algorithm.

2.2.2. Task Classifier Design

The typical tasks of MA include cruise, ascent, descent, acceleration and deceleration. According to the command signal, the task phase of the aircraft can be accurately distinguished. However, in the actual flight control, the speed and altitude of the aircraft will not always equal the command signal. For example, the altitude of the aircraft will have a process of adjustment in the ascent phase. There will be oscillation before reaching the stable state and the speed will change accordingly. At this time, it is difficult to judge whether the aircraft is in the ascent phase according to the increase in altitude. Therefore, a task classifier is needed to distinguish different task types according to the flight state of the aircraft in different task phases. However, the state of the aircraft at a moment cannot well represent its task phase, so the state of a time series is needed as the input of the task classifier. Considering that LSTM has a good effect in dealing with the problem of time series classification, LSTM is used to design the task classifier.

The LSTM network is an improved version of a recurrent neural network that can effectively deal with the prediction or classification problem of sequence data. The data of each moment will be stored in the memory unit. It has three gating devices to determine which data should be kept and which irrelevant data should be discarded in time. The memory unit structure of LSTM is shown in Figure 3.

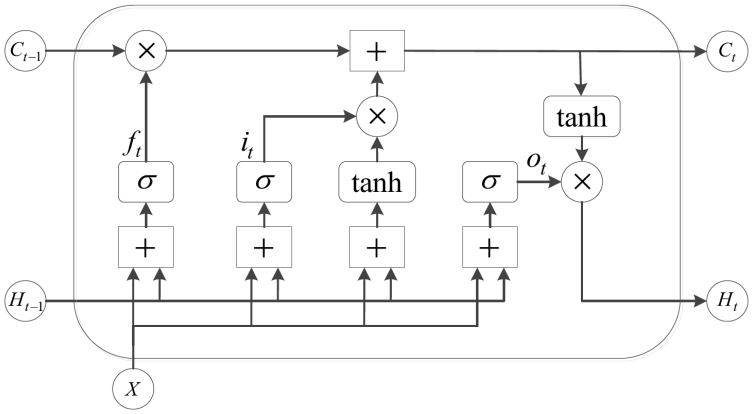

Figure 3. The memory unit structure of LSTM.

The input of the memory unit includes the input value X of the current network, the output value H_{t-1} at the previous time and the unit state C_{t-1} at the previous time. There are three gates in the memory unit, which are the input gate, forget gate and output gate. X first judges whether the input is valid through the input gate, then judges whether

to choose to forget the data stored in the memory unit through the forget gate, and finally judges whether to output the data at this moment through the output gate. The equations of the three gates are as follows.

$$\begin{cases} i_t = \sigma(W_t \cdot [H_{t-1}, X] + b_t) \\ \tilde{c}_t = \tanh(W_e \cdot [H_{t-1}, X] + b_e) \\ f_t = \sigma\left(W_f \cdot [H_{t-1}, X] + b_f\right) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\ o_t = \sigma(W_o \cdot [H_{t-1}, X] + b_o) \\ H_t = o_t \cdot \tanh(c_t) \end{cases} \tag{8}$$

where W is the parameter matrix of the memory unit, b is the bias of the memory unit, σ and tanh are activation functions.

Let the state space of the algorithm be $s = [h_r,\ V_r,\ \mathrm{d}h_r,\ \mathrm{d}V_r,\ h_c,\ V_c,\ \mathrm{d}h_c,\ \mathrm{d}V_c]$, where h_r and V_r are the actual altitude and actual speed of the aircraft, h_c and V_c are the command altitude and command speed, $\mathrm{d}h_c$ and dV_c are the variation of the command altitude and command speed, $\mathrm{d}h_r$ and $\mathrm{d}V_r$ are the variation of the actual altitude and actual speed. Before the aircraft state is input to the task classifier, the data sampling process is performed. The sampling interval is 1s and the flight state data within 5s is saved. Then, the data input to the task classifier at a certain time t is $[s_{t-4},\ s_{t-3},\ s_{t-2},\ s_{t-1},\ s_t]^{\mathrm{T}}$. LSTM classifies the task phase of the aircraft at time t according to the actual value, command value and variation of the aircraft speed and altitude in this period. After repeated experiments, the task classifier is designed as shown in Figure 4.

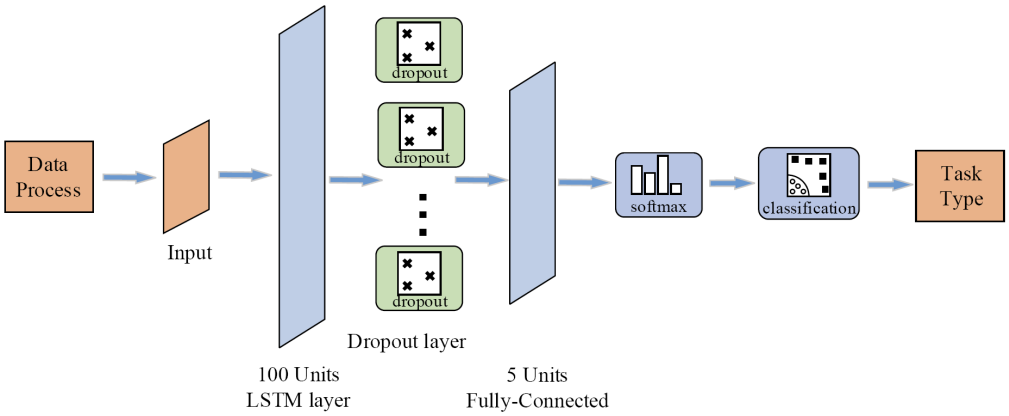

Figure 4. LSTM network.

Among them, the data process module saves the output state data of MA, inputs the data to the input layer of the LSTM network when receiving the data for five time periods, and maintains the data length for five time periods during the simulation. The data passes through the LSTM layer from the input layer. After debugging, the LSTM layer is set to 100 neurons. To prevent over fitting, the dropout layer is added before the full connection layer. Finally, one of the categories 1–5 is output through the classification layer. The corresponding relationship between the output result and the task type is [1: ascent, 2: descent, 3: acceleration, 4: deceleration, 5: cruise].

2.2.3. Reward Function Design

The reward function can be designed according to the task type after the task classifier has derived the task phase of the aircraft. Referring to [32], the aircraft needs to obtain the maximum ratio of lift to drag and the minimum fuel consumption during the cruise phase, which can extend the endurance time. In the ascent phase, the aircraft needs to obtain the maximum lift to improve the ascent speed. Similarly, the aircraft needs the least lift

in the descent phase. In the acceleration phase, the drag needs to be reduced to improve the maneuverability. In the deceleration phase, the drag needs to be increased. Therefore, the lift and drag needed in each task phase can be used as the design index of reward function. When the wing configuration of MA is changed, its lift and drag will also change. The lift and drag can also be taken as the state output of the aircraft consequently, then integrated with fuel consumption and tracking error and other indicators. The reward function can be designed as follows.

$$R = -t(|(h_r - h_c)| + |(V_r - V_c)|) + R_T \tag{9}$$

where t is the current time and R_T is the reward corresponding to different task types, defined as follows.

$$R_T = \begin{cases} L\ if\ T = 1 \\ 1/L\ \ if\ T = 2 \\ 1/D\ \ if\ T = 3 \\ D\ if\ T = 4 \\ L/D\ + \delta_T\ if\ T = 5 \end{cases} \tag{10}$$

where δ_T is the throttle opening, T is the task type output by the task classifier, L is the lift of the aircraft, and D is the flight resistance.

When calculating the reward function, it is necessary to normalize the indicators to make the data scale to [−10,10], so that each indicator has the same degree of influence on the reward function. At the same time, it also prevents certain data from being too large to cause the agent unable to explore the action space comprehensively.

2.2.4. The Process of the DDPGwTC Algorithm

The basic structure of the DDPGwTC algorithm is shown in Figure 5.

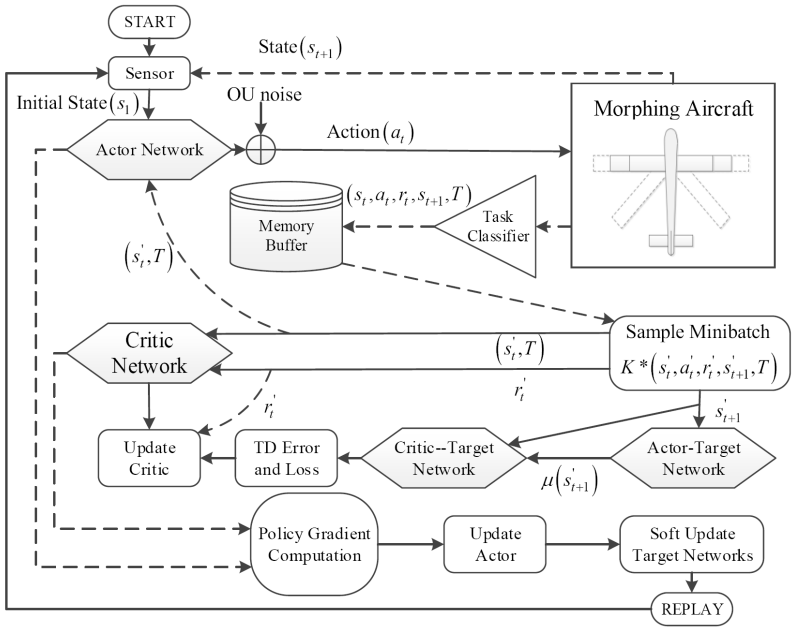

Figure 5. The structure of the DDPGwTC algorithm.

Among them, the flight state of MA is transformed into a task signal by the task classifier and input into the policy network and value network, adding "prior knowledge" to the network. In the algorithm, the idea of experience replay and target network of the DDPG algorithm are still used. First, the flight data of MA are generated in the same environment, so these data do not meet the condition of being independently identically

distributed. To break the correlation between the data, it is necessary to sample the data randomly. At the same time, to increase the efficiency of training, it is necessary to perform the minibatch learning. Therefore, an experienced replay buffer R is created to store the data generated by the interaction between the agent and the environment, that is, state, action and reward. The experience replay buffer is a container of limited size. The interactive data (s_t, a_t, r_t, s_{t+1}) are sampled from the environment according to the action policy and stored in the experience replay buffer. When the experience replay buffer is full, the earliest data are discarded, and a small batch of interactive data is randomly sampled from the experience replay buffer at each simulation step to update the policy network and value network. The features of the four networks in the algorithm are as follows. The actor network is responsible for updating network parameter θ^{μ}, and selecting action a to interact with the environment according to current state s. The generated interactive data are stored in the experience replay buffer. The actor–target network is responsible for sampling the next state s' from the experience replay buffer and selecting action a'. The critic network is responsible for updating the network parameter θ^{Q} and calculating the current Q value. The critic–target Network is responsible for calculating the value of Q'. The parameters of the target network are updated slowly with the current network in a soft update approach to ensure that the calculated target value is relatively stable. The update approach is as follows.

$$\begin{cases} \theta^{\mu'} = \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \\ \theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'} \end{cases} \tag{11}$$

where τ is the update coefficient, generally taken as 0.01.

In summary, the algorithm flow of the DDPGwTC is as follows.

Algorithm 1 Deep deterministic policy gradient with task classifier (DDPGwTC)

Randomly initialize critic $Q\left(s, a\middle|\theta^{Q}\right)$ and actor $\mu(s|\theta^{\mu})$ neural networks with weights $\theta^{Q}$and θ^{μ}.
Initialize target network Q' and $\mu'$with weights $\theta^{Q'}$=θ^{Q}, $\theta^{\mu'}$=θ^{μ}.
Initialize replay buffer R .
for episode= $1, \cdots, N$ **do**
  Initialize a random process $\mathcal{N}$ for action exploration.
  Receive initial observation state s_1.
  for episode= $1, \cdots, M$ **do**
    Select action $a_t = \mu(s_t|\theta^{u}) + \mathcal{N}$ according to the current policy and exploration noise.
    Execute action a_t and observe new state s_{t+1}.
    Classify s_{t+1} to get the task signal T.
    Get reward r_t according to T.
    Store transition $(s_t, a_t, r_t, s_{t+1}, T)$ in experience replay buffer R.
    Sample a random minibatch of K transitions $(s_t, a_t, r_t, s_{t+1}, T)$ from R.
    Set y_i=$r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{Q'}\right)$.
    Update critic by minimizing the loss L:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q\left(s_i, a_i\middle|\theta^{Q}\right)\right)^2.$$

    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{M}\sum_{i=1}^{M}\nabla_a Q\left(s, a\middle|\theta^{Q}\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s_t};$$

17: Update target networks:

$$\begin{cases} \theta^{\mu'} = \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \\ \theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'} \end{cases};$$

18: **end for**
19: **end for**

Remark 1. *Compared with DDPG algorithm, the DDPGwTC algorithm classifies state s_{t+1} to obtain task signal T, then obtains reward r_t according to T. By using the task classifier, the convergence speed of the algorithm can be improved and the reward value is more stable in subsequent iterations.*

3. Network Training

3.1. Task Classifier Network Training

Before training the DDPGwTC algorithm, the task classifier needs to be pre-trained so that it can classify the task phase of the aircraft when training the DDPGwTC algorithm. Through the simulation of the MA system, 1000 pieces of flight state data of the aircraft are obtained as training samples, including the data of each task phase, the data at different altitudes and speeds, and the data under changes of different wingspan and sweep angle. First, the data are divided into the training set, validation set and test set according to the ratio of 6:2:2. After repeated debugging, the batch size is set to 60, the learning rate is set to 0.1, and the training period is set to 100. Finally, the result is shown in Figure 6, and an accuracy of 94.32% can be achieved on the test set.

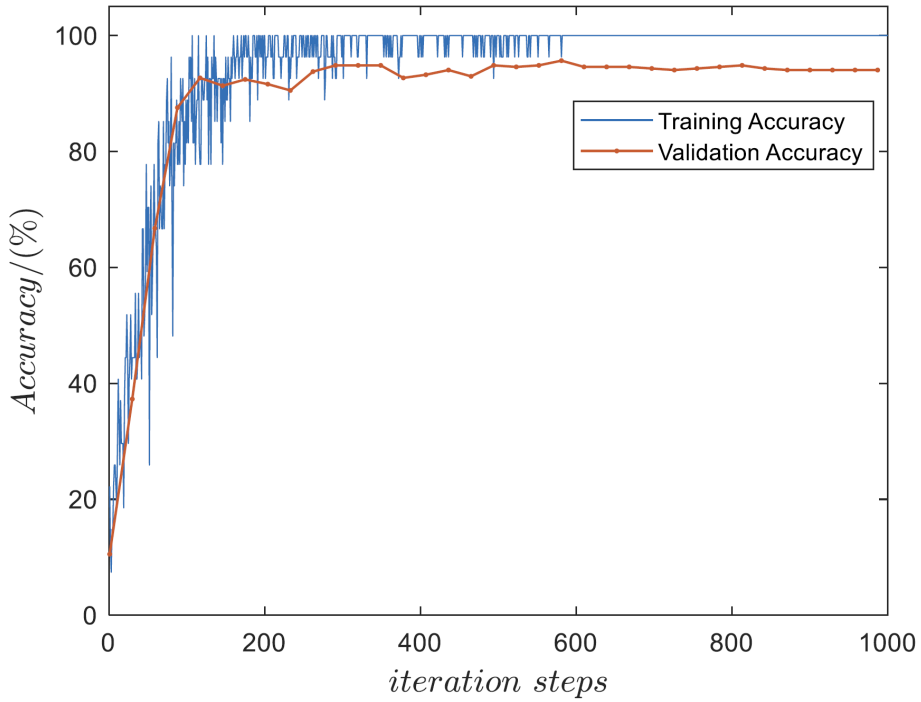

Figure 6. Task classifier network training result.

3.2. DDPGwTC Algorithm Training

The T-S fuzzy control system of MA is used as the environment model and the algorithm training is carried out based on the MATLAB RL Toolbox software platform. The hardware adopts NVIDIA GeForce RTX 2070 GPU. The training process is as follows. At the beginning of each round of training, a flight trajectory with a duration of 400 s is set randomly. Then, MA flies according to the preset flight trajectory. During this period, the agent adjusts the morphing strategy according to the flight phase of the aircraft. Finally,

the flight task until the aircraft completes this part of the trajectory is regarded as the end of a round of training and the cumulative return is obtained. Referring to the experience of parameter adjustment in [25], the hyperparameters of the DDPGwTC algorithm are set as shown in Table 1 after repeated debugging, and the structures of the actor network and critical network are shown in Figures 7 and 8.

Table 1. Hyperparameter values used for DDPGwTC algorithm.

Hyperparameter	Value
Target Update Factor /τ	0.001
Actor Learning Rate	0.0005
Critic Learning Rate	0.001
Experience Replay Buffer Capacity	10^6
Minibatch Size	128
Discount Factor /γ	0.99
Maximum Number of Scenes	200

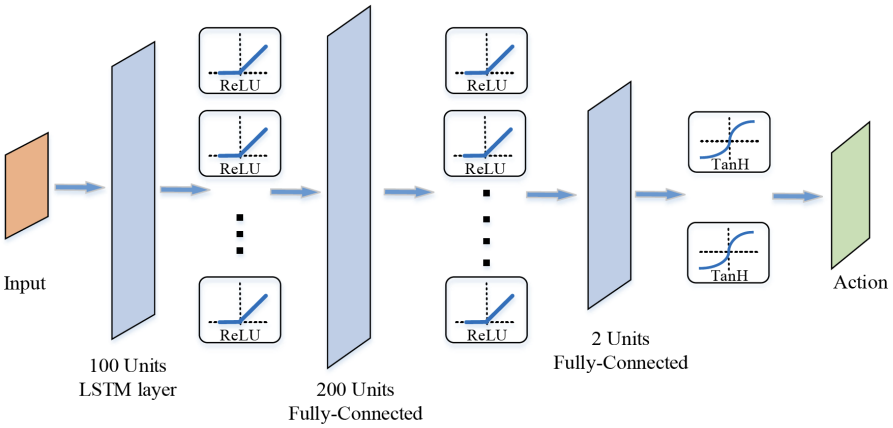

Figure 7. The structure of actor network.

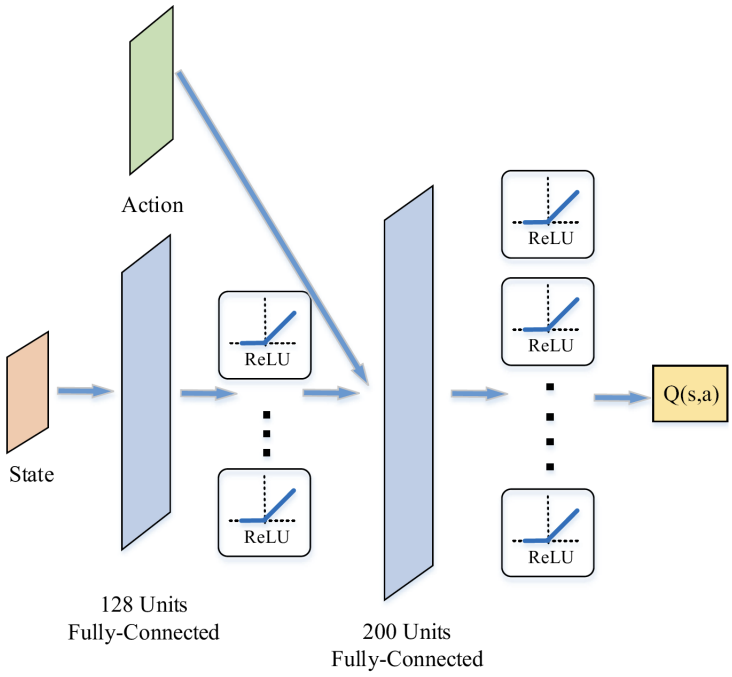

Figure 8. The structure of critic network.

After 200 rounds of training, the cumulative return of each round of the DDPG algorithm and the DDPGwTC algorithm is shown in Figure 9. It can be seen from the figure that the DDPG algorithm without a task classifier converges slowly and tends to fall into local optimal value. The DDPGwTC algorithm converges quickly and converges to a higher reward value policy after 80 iterations, and the reward is more stable in the later iterations.

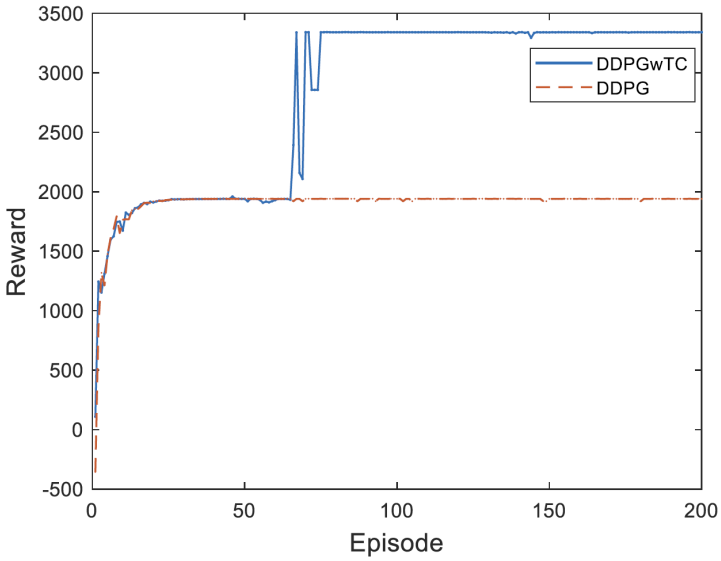

Figure 9. The training result of the DDPGwTC algorithm.

4. Simulation Analysis

To verify the effect of the DDPGwTC algorithm in the shape decision making of MA, the trained agent is used as the shape controller of the aircraft. The flight trajectory is set as follows. Initially, the aircraft cruises at 2000 m altitude with a speed of 34 m/s and sets the speed to 50 m/s at 50 s, sets the speed to 30 m/s at 120 s, sets the altitude to 2050 m at 170 s, sets the altitude to 2000 m at 280 s, and finally maintains the cruise state until the end of 400 s. The altitude and speed command curves are shown in Figure 10. The agent which is obtained by using the trained DDPGwTC algorithm is applied to the shape decision-making algorithm framework of MA as shown in Figure 2 for simulation flight, and the speed and altitude tracking curves obtained are shown in Figures 11 and 12. The output curve of the task classifier and reward value during the flight are shown in Figures 13 and 14. The morphing rate curve of MA is shown in Figure 15, and the corresponding shape diagram is marked in the flight time period shown in Figure 10 as the form of a small aircraft icon.

It can be seen from Figure 13 that the task type output by the classifier is basically consistent with the preset trajectory, which indicates that the task classifier can accurately classify the task phase of MA according to its flight state. Figure 14 shows that when the task type changes, the agent will take corresponding action to maximize the reward value obtained by the aircraft. Figure 15 shows the morphing strategy adopted by the agent in the flight process. During the cruise, ascent and deceleration phases of MA, the agent controls the aircraft to extend wings and increase the sweep angle. During the acceleration and descent phases of MA, the agent controls the aircraft to shrink wings and reduce the sweep angle. According to the expert experience, it is known that the morphing strategy adopted by the agent in each task phase of the flight is optimal. Therefore, it can be seen from the above simulation results that the agent based on the DDPGwTC algorithm enables MA change to the optimal shape in different task environments.

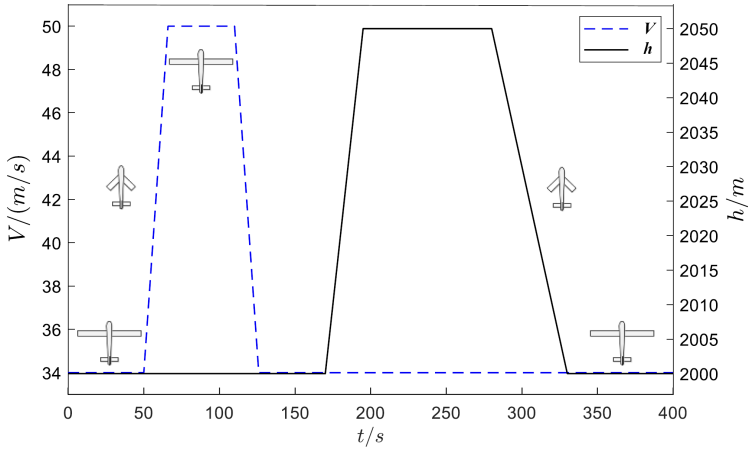

Figure 10. Flight trajectory curve.

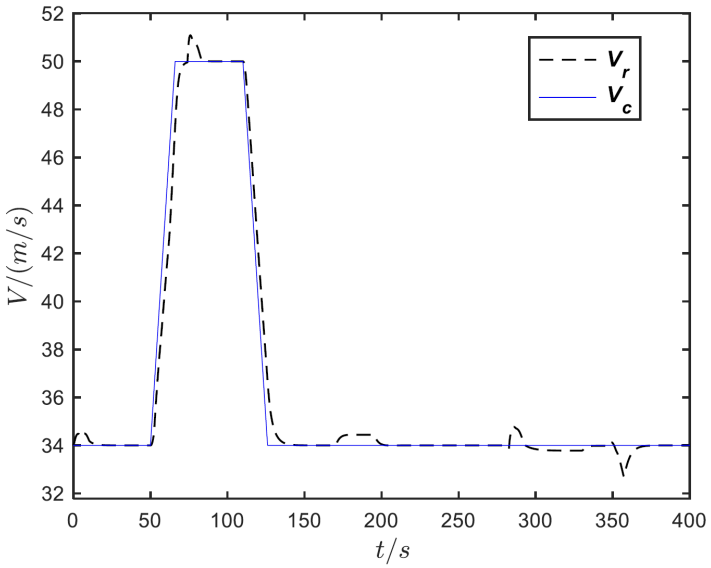

Figure 11. Velocity tracking curve.

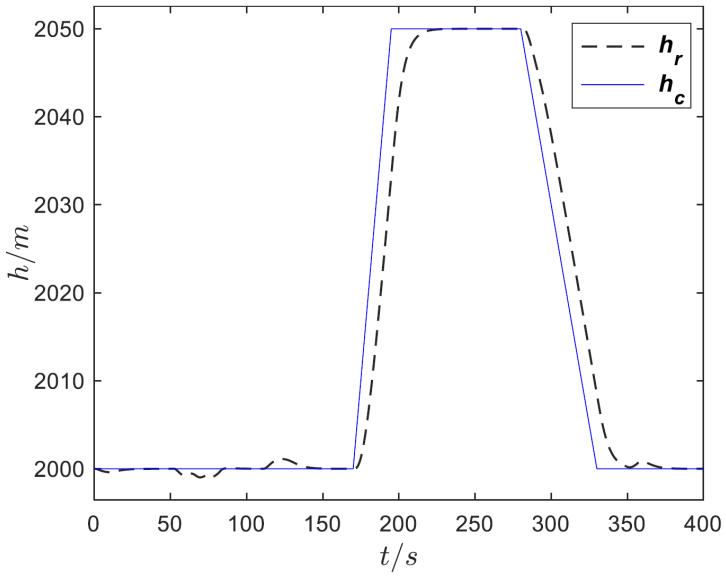

Figure 12. Height tracking curve.

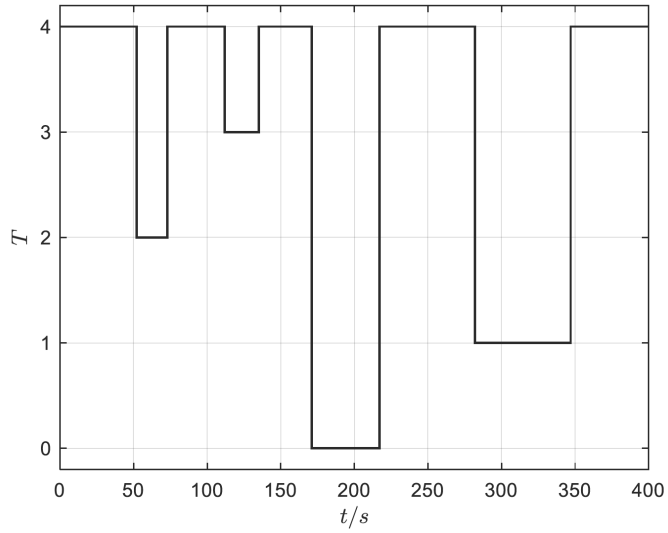

Figure 13. Task classifier output curve.

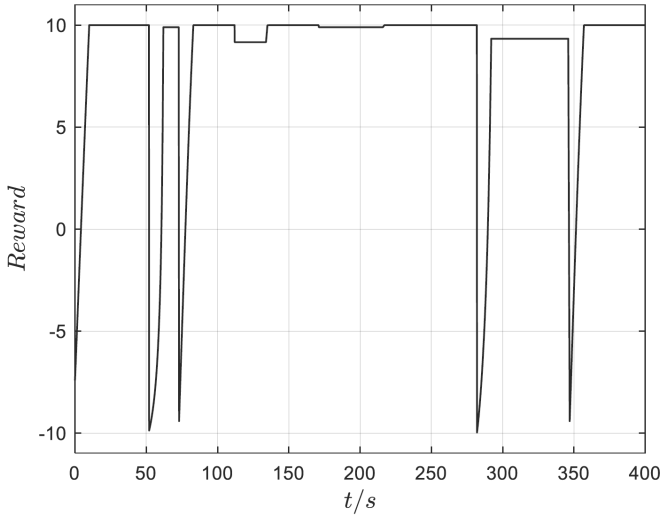

Figure 14. Reward curve.

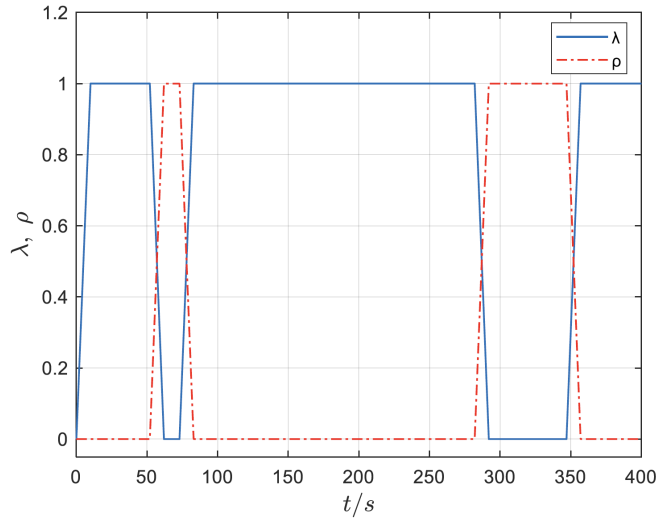

Figure 15. Morphing rate curve.

5. Conclusions

In this article, we propose an algorithm framework for MA shape autonomous decision making. Firstly, the DDPG algorithm is used to provide important information for shape decision-making. Secondly, the DDPGwTC algorithm based on the LSTM network is proposed to improve the convergence speed of the DDPG. Finally, flight simulation is performed on the trained network using a T-S fuzzy controller. After simulation experiments, it was found that the proposed algorithm converges faster than the DDPG algorithm, and the trained intelligent agent can guide the aircraft to make optimal shape decisions in different task environments, improving the intelligence and environmental adaptability of the aircraft.

This article uses a T-S fuzzy controller, but the selected controller is relatively simple. Therefore, in subsequent research, more advanced controllers need to be combined, such as switching LPV control, adaptive control, etc., to improve the tracking effect and stability of the variant aircraft during the shape change process.

Author Contributions: Conceptualization and methodology, W.J. and C.Z.; software, W.J.; validation, W.J., C.Z. and K.W.; investigation, D.H.; data curation, C.Z.; writing, W.J. and C.Z.; supervision, Y.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for Key Research and Development Programs of Jiangsu grant number BE2020082-1 and the General Project of Natural Science Foundation of Hunan Province grant number 2022JJ30162.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Kangsheng Wu was employed by the company Zoomlion Heavy Industry Science and Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Weisshaar, T.A. Morphing aircraft systems: Historical perspectives and future challenges. *J. Aircraft* **2013**, *50*, 337–353. [CrossRef]
2. Ajaj, R.M.; Parancheerivilakkathil, M.S.; Amoozgar, M.; Friswell, M.I.; Cantwell, W.J. Recent developments in the aeroelasticity of morphing aircraft. *Prog. Aerosp. Sci.* **2021**, *120*, 100682. [CrossRef]
3. Wang, Q.; Gong, L.; Dong, C.; Zhong, K. Morphing aircraft control based on switched nonlinear systems and adaptive dynamic programming . *Aerosp. Sci. Technol.* **2019**, *93*, 105325. [CrossRef]
4. Li, W.; Wang, W.; Huang, W. Morphing aircraft systems: Historical perspectives and future challenges. *Appl. Sci.* **2021**, *11*, 2505.
5. Li, R.; Wang, Q.; Dong, C. Morphing Strategy Design for UAV based on Prioritized Sweeping Reinforcement Learning. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 2786–2791.
6. Zhang, J.; Shaw, A. Aeroelastic model and analysis of an active camber morphing wing. *Aerosp. Sci. Technol.* **2021**, *111*, 106534. [CrossRef]
7. Grigorie, T.L.; Botez, R.M. A Self–Tuning Intelligent Controller for a Smart Actuation Mechanism of a Morphing Wing Based on Shape Memory Alloys. *Actuators* **2023**, *12*, 350. [CrossRef]
8. Huang, J.; Fu, X.; Jing Z. Singular dynamics for morphing aircraft switching on the velocity boundary. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *95*, 105625. [CrossRef]
9. Burdette, D.A.; Martins, J.R.R.A. Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization. *Aerosp. Sci. Technol.* **2018**, *81*, 192–203. [CrossRef]
10. Li, W.; Wang, W.; Huang, X.; Zhang, S.; Li, C. Roll Control of Morphing Aircraft with Synthetic Jet Actuators at a High Angle of Attack. *Appl. Sci.* **2021**, *11*, 505. [CrossRef]
11. Yan, B.; Dai, P.; Liu, R. Adaptive super-twisting sliding mode control of variable sweep morphing aircraft. *Aerosp. Sci. Technol.* **2019**, *92*, 198–210. [CrossRef]
12. Jiang, W.; Wu, K.; Wang, Z. Gain-scheduled control for morphing aircraft via switching polytopic linear parameter-varying systems. *Aerosp. Sci. Technol.* **2020**, *107*, 106242. [CrossRef]
13. Cheng, L.; Li, Y.; Yuan, J.; Ai, J.; Dong, Y. $\mathcal{L}_1$ Adaptive Control Based on Dynamic Inversion for Morphing Aircraft. *Aerospace* **2023**, *10*, 786. [CrossRef]
14. Mnih, V.; Kavukcuoglu, K.; Silver, D. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529—533. [CrossRef]

15. Wang, Z.; Schaul, T.; Hessel, M. Dueling network architectures for deep reinforcement learning. *Int. Conf. Mach. Learn.* **2016**, *48*, 1995–2003.
16. Hausknecht, M.; Stone, P. Deep recurrent q-learning for partially observable MDPs. In Proceegings of the Association for the Advancement of Artificial Intelligence Presented the 2015 Fall Symposium Series, Arlington, VA, USA, 12–14 November 2015; Volume 1507, p. 06527.
17. Sutton, R.S.; McAllester, D.A.; Singh, S.P. Policy gradient methods for reinforcement learning with function approximation. *Neural Inf. Process. Syst.* **1999**, *12*, 1057–1063.
18. Silver, D.; Lever, G.; Heess, N. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014; pp. 387–395.
19. Valasek, J.; Tandale, M.D.; Rong, J. A reinforcement learning - adaptive control architecture for morphing. *J. Aerosp. Comput. Inf. Commun.* **2005**, *2*, 174–195. [CrossRef]
20. Valasek, J.; Doebbler, J.; Tandale, M.D. Improved adaptive-reinforcement learning control for morphing unmanned air vehicles. *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)* **2008**, *38*, 1014–1020. [CrossRef] [PubMed]
21. Lampton, A.; Niksch, A.; Valasek, J. Morphing airfoils with four morphing parameters. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008; pp. 72–82.
22. Lampton, A.; Niksch, A.; Valasek, J. Reinforcement learning of a morphing airfoil-policy and discrete learning analysis. *J. Aerosp. Comput. Inf. Commun.* **2010**, *7*, 241—260. [CrossRef]
23. Lampton, A.; Niksch, A.; Valasek, J. Reinforcement learning of morphing airfoils with aerodynamic and structural effects. *J. Aerosp. Comput. Inf. Commun.* **2015**, *6*, 30—50. [CrossRef]
24. Yan, B.; Li, Y.; Dai, P. Adaptive wing morphing strategy and flight control method of a morphing aircraft based on reinforcement learning. *J. Northwest. Polytech. Univ.* **2019**, *37*, 656–663. [CrossRef]
25. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
26. Wen, N.; Liu, Z.; Zhu, L. Deep reinforcement learning and its application on autonomous shape optimization for morphing. *J. Astronaut.* **2017**, *38*, 1153–1159.
27. Goecks, V.G.; Leal, P.B.; White, T. Control of morphing wing shapes with deep reinforcement learning. In Proceedings of the 2018 AIAA Information Systems—AIAA Infotech@ Aerospace, Kissimmee, FL, USA, 8–12 January 2018; p. 2139.
28. Xu, D.; Hui, Z.; Liu, Y.; Chen, G. Morphing control of a new bionic morphing UAV with deep reinforcement learning. *Aerosp. Sci. Technol.* **2019**, *92*, 232–243. [CrossRef]
29. Watkins, C.J.C.H. Learning from Delayed Rewards. Ph.D. Thesis, King's College, London, UK, 1989 .
30. Jiang, W.; Wu, K.; Bao, C.; Xi, T. T-S Fuzzy Modeling and Tracking Control of Morphing Aircraft. *Lect. Notes Electr. Eng.* **2022**, *644*, 2869–2881 .
31. Shen, X.; Dong, C.; Jiang, W. Longitudinal control of morphing aircraft based on T-S fuzzy model. In Proceedings of the IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014.
32. Seigler, T.M. Dynamics and Control of Morphing Aircraft. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2005.