Path Following Control for Underactuated Airships with Magnitude and Rate Saturation

Huabei Gou 1*, Xiao Guo 2, Wenjie Lou 3, Jiajun Ou 1 and Jiace Yuan 1*

1 School of Aeronautic Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China; ghbuaaa@buaa.edu.cn (H.G.); oujiajun@buaa.edu.cn (J.O.);
yuanjiace@buaa.edu.cn (J.Y.)
2 Frontier Institute of Science and Technology Innovation, Beijing University of Aeronautics and Astronautics, Beijing 100191, China; guoxiao@buaa.edu.cn
3 School of Electronic and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
* Correspondence: louwenjie@buaa.edu.cn

Received: 15 October 2020; Accepted: 10 December 2020; Published: 15 December 2020

Abstract: This paper proposes a reinforcement learning (RL) based path following strategy for underactuated airships with magnitude and rate saturation. The Markov decision process (MDP) model for the control problem is established. Then an error bounded line-of-sight (LOS) guidance law is investigated to restrain the state space. Subsequently, a proximal policy optimization (PPO) algorithm is employed to approximate the optimal action policy through trial and error. Since the optimal action policy is generated from the action space, the magnitude and rate saturation can be avoided. The simulation results, involving circular, general, broken-line, and anti-wind path following tasks, demonstrate that the proposed control scheme can transfer to new tasks without adaptation, and possesses satisfying real-time performance and robustness.

Keywords: reinforcement learning; path following; underactuated airships; magnitude and rate saturation

1. Introduction

As a kind of lighter-than-air vehicles, airships have distinct advantages over other vehicles in ultra-long duration flight, low fuel consumption, making them a cost-effective platform for communication relay, monitoring, surveillance, and scientific exploration. Many countries regard airships as one of the most important platforms in near space and have developed techniques about airships for decades [1–3].

With the rapid progress of airship technologies, the essential role of flight control is more prominent. Path following is one of the most frequently performed tasks in flight control, which requires the airship to follow a predefined geometric path without temporal constraint. Due to the inherent dynamics nonlinearity, unknown dynamics, parametric uncertainty, and external disturbance, airship path following control becomes a challenging research topic [4,5]. Numerous achievements on the path following problems have been witnessed, employing various methods such as sliding mode control (SMC) [6,7], H-infinity [8,9], backstepping control [4,10–12], fuzzy control [13–18], and so forth. In Reference [19], a computationally efficient observer-based model predictive control (MPC) method is presented to achieve arbitrary path following for marine vessels. In Reference [20], a path following controller for a quadrotor with a cable suspended is proposed. The coordinated path following control for multiple nonlinear autonomous vehicles connected by digital networks is studied in Reference [21]. In Reference [22], the authors investigate the reliable H-infinity path following controller for an autonomous ground vehicle.
Although various studies about path following control are investigated, there are still many challenges and obstacles for viable implementation. One of the challenges is the actuators’ magnitude and rate saturation, which is ubiquitous in engineering. For airships, the magnitude saturation is the upper and lower limits of the thrust and the rudder deflection, while the rate saturation is the corresponding acceleration saturation. The saturation problem normally means the amplitude and rate constraints of the input signals during the controller design. Once the constraints are violated, it may lead to performance deterioration or system instability [23,24]. Various control strategies are investigated to handle the saturation problem [25–29]. In Reference [30], a backstepping controller combined with auxiliary systems is designed to meet the input magnitude and rate constraints for a two-blade propeller system of an aircraft. In Reference [31], a decentralized state feedback controller based on the linear matrix inequality (LMI) conditions is adopted to a class of nonlinear interconnected systems subject to the magnitude and rate constraints. In Reference [32], the LMI and Lyapunov direct techniques are combined to deal with the input magnitude and rate saturation for a flexible spacecraft. In Reference [33], the backstepping scheme with smooth hyperbolic tangent function is proposed to address the magnitude and rate saturation for the nonlinear three-dimensional Euler-Bernoulli beam. However, the problem has not been fully studied in the domain of airship path following control. Due to the complexity of the problem, it is difficult to implement Lyapunov based methods. MPC is served as an effective solution by considering control problem as a process of optimization, but heavy computing burden hampers the real-time implementation.

Recently, reinforcement learning (RL) has been successfully applied in robot control, underwater vehicle guidance, and path planning [34–36]. Owing to the performance of model insensitivity, RL can achieve control objectives for various complex dynamic systems. Besides, RL becomes so compelling for its prominent advantage in treating optimization problems. Therefore, it is feasible to solve the airships’ magnitude and rate saturation problems from the perspective of optimization with RL strategies. Generally, RL methods can be divided into two categories—policy search methods and value-based methods. For the policy search methods, the action-selection policies are explicit rather than implicitly derived from separate value functions, which makes it more efficient than value-based methods when dealing with the continued-states and high-dimension problems [37,38]. There are several popular RL algorithms such as trust region policy optimization (TRPO) [39], proximal policy optimization (PPO) [40], soft actor-critic (SAC) [41], and twin delayed deep deterministic policy gradient (TD3) [42]. Considering the efficiency and stability, the PPO algorithm is adopted in our approach, which is one of the state-of-the-art learning algorithms.

With the enrichment of application scenarios, the controller of airships must have the ability to adapt to a variety of tasks. There are two challenges that must be considered during the controller design process [43]. First, transferring policies proves difficult on task distributions for the optimal policy varies dramatically across tasks. Second, transferring policies affects the time and sample efficiency in the adaptation phase. Motivated by the discussions mentioned above, the RL-based path following controller is proposed for underactuated airships subject to magnitude and rate saturation in this paper. To realize real-time performance and transfer action policy between tasks, an error bounded LOS guidance law is presented to restrain the state space, while a PPO-based algorithm is employed to acquire the action policy through trial and error. Frequently performed tasks are converted to a point-following task while the point is always distributed in a specified bounded space. Exhaustive training in this space will generate optimal action policies.

Contributions of this paper are as follows:

- An RL-based path following controller is proposed for underactuated airships with magnitude and rate saturation. Compared with the nonlinear MPC method referenced from Reference [44], the proposed method has better performance in real-time and comparable control quality.
- An error bounded LOS guidance law is presented to provide task-independent state space, enabling the trained controller to transfer to new tasks without adaptation. Besides, a PPO-based algorithm is employed to generate the action policy through trial and error.
• The capability of transferring between tasks is demonstrated by performing circular, general, and broken-line path following tasks, while the robustness is proved by circular path following under wind disturbance.

The rest of this paper is organized as follows: Section 2 presents the airship model. The design of the controller is shown in Section 3. Section 4 simulates the proposed controller. Discussion and conclusion are given in Sections 5 and 6, respectively.

2. Airship Model

The airship studied in this paper is depicted in Figure 1. The static lift of the airship is provided by the helium contained in the envelope. The aerodynamic control surfaces attached to the rear of the airship, namely the tails and rudders, offer the control torques for the airship and enhance the course stability. The propellers are mounted symmetrically on the sidewall of the gondola fixed at the bottom of the envelope.

Figure 1. Structure and coordinate systems of the airship.

To develop the airship model, the body reference frame (BRF) is firstly defined. The BRF is attached to the airship with its origin \( O \) coincident with the center of volume. \( O_x \) points towards the head of the airship, \( O_y \) is perpendicular to \( O_x \) towards the right of the airship, and \( O_z \) is determined by the right-hand rule. The origin \( O_x \) of the earth reference frame (ERF) is fixed on the ground, \( O_y \) and \( O_z \) point to the north and east, respectively. \( O_y \) is determined by the right-hand rule.

According to Reference [45], the model of the airship is described as:

\[
\begin{bmatrix}
\dot{\mathbf{\zeta}} \\
\dot{\mathbf{\Theta}}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{K} & \mathbf{0}_{3 \times 3} \\
\mathbf{0}_{3 \times 3} & \mathbf{R}
\end{bmatrix}
\begin{bmatrix}
\mathbf{v} \\
\mathbf{\Omega}
\end{bmatrix},
\]

(1)

\[
\bar{\mathbf{A}}
\begin{bmatrix}
\mathbf{v} \\
\mathbf{\Omega}
\end{bmatrix} = \mathbf{N} + \mathbf{G} + \mathbf{B}
\begin{bmatrix}
\mathbf{u}_f \\
\mathbf{u}_\delta
\end{bmatrix},
\]

(2)

where \( \mathbf{\zeta} = [x, y, z]^T \) and \( \mathbf{\Theta} = [\phi, \theta, \psi]^T \) are the position and attitude of airship described in ERF, respectively. \( \mathbf{v} = [u, v, w]^T \) and \( \mathbf{\Omega} = [p, q, r]^T \) are velocity and angular velocity described in BRF, respectively. The detailed expressions of \( \bar{\mathbf{A}}, \mathbf{N}, \bar{\mathbf{G}}, \mathbf{B}, \mathbf{u}_f \) and \( \mathbf{u}_\delta \) are described in Reference [45].

To facilitate the design of the controller, the attitude motion of pitch and roll in the vertical plane is ignored and only the horizontal motion is considered in this study [9]. Thus, the values of the variables
where \( w, p, q, \phi, \theta, \text{ and } z \) are set as zero. Consequently, the planar linear kinematics and dynamic state-space equations can be derived from (1) and (2) as:

\[
\begin{bmatrix}
\dot{\psi} \\
\dot{x} \\
\dot{y}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \psi & -\sin \psi \\
0 & \sin \psi & \cos \psi
\end{bmatrix}
\begin{bmatrix}
r \\
u \\
v
\end{bmatrix}
\]

\( X_1 = AX_1 + BF \)
\( X_2 = CX_1 + DF \)

3. Path Following Controller Design

In this section, a path following controller is developed for underactuated airships with magnitude and rate saturation. The proposed controller consists of two sub-systems: error bounded LOS guidance and PPO control. The guidance loop provides the desired heading angle and the target position, and PPO calculates the near-optimal solution in bounded space. In PPO control, the Markov decision process (MDP) model is constructed firstly. Then, the reward function is designed based on the MDP model. Finally, the optimization process of PPO is introduced. Figure 2 shows the structure of the controller.

During the design process, the state space is reduced to an acceptable range. After adequate training, most states in the space will be experienced. Consequently, the desired action policy can be obtained, realizing the real-time path following.

3.1. Error Bounded LOS Guidance

The underactuated airship is desired to follow a continuous path \( \zeta_p(\omega) = [x_p(\omega), y_p(\omega)]^T \) parameterized by a time-independent variable \( \omega \). Suppose that the target position provided by the path is \( \zeta_p = (x_p, y_p) \), a path parallel frame (PPF) can be defined: its origin point is \( \zeta_p, \zeta_p x_p \) is parallel
with the local tangent vector, and \( \zeta_{py} \) is the normal vector pointing to the right. The ERF should be positively rotated by an angle \( \psi_p \) to reach PPF.

\[
\psi_p = \arctan2\left(\frac{dy_p}{d\omega}, \frac{dx_p}{d\omega}\right).
\]

Thus, the tracking error in PPF can be expressed as follows:

\[
e = [s, e]^T = R(\psi_p)(\zeta - \zeta_p),
\]

where \( s \) is the along-track error, \( e \) is the cross-track error, and \( R(\psi_p) \) is given by:

\[
R(\psi_p) = \begin{bmatrix} \cos \psi_p & \sin \psi_p \\ -\sin \psi_p & \cos \psi_p \end{bmatrix}.
\]

To reduce the state space, the target position \( \zeta_c = (x_c, y_c) \) is selected as follows:

\[
\zeta_c = \zeta + K_r \tanh\left(\frac{\zeta_p - \zeta}{K_r}\right),
\]

where \( K_r > 0 \). The target position \( \zeta_c \) will remain in a circle near current position.

\[
\zeta - \zeta_c = K_r \tanh\left(\frac{\zeta - \zeta_p}{K_r}\right).
\]

\[
K_r \tanh\left(\frac{\zeta - \zeta_c}{K_r}\right) = \zeta - \zeta_p.
\]

Obviously, the tracking error \( \zeta - \zeta_p \) will converge to zero as \( \zeta \) and \( \zeta_c \) converge to zero. Considering Lyapunov function as follows:

\[
V_e = \frac{1}{2} K_r^2 (R(\psi_p)\tanh(\frac{\zeta_c - \zeta}{K_r}))^T R(\psi_p)\tanh(\frac{\zeta_c - \zeta}{K_r}).
\]

Substituting Equation (10) into Equation (11):

\[
V_e = \frac{1}{2} K_r^2 (R(\psi_p)\tanh(\frac{\zeta - \zeta_p}{K_r}))^T R(\psi_p)\tanh(\frac{\zeta - \zeta_p}{K_r})
\]

\[
= \frac{1}{2} (R(\psi_p)(\zeta - \zeta_p))^T (R(\psi_p)(\zeta - \zeta_p))
\]

\[
= \frac{1}{2} e^T e.
\]

Differentiating Equation (12) with respect to time, it leads to:

\[
V_e = e^T \dot{e}
= e^T (R(\psi_p)(\zeta - \zeta_p) + R(\psi_p)(\zeta - \zeta_p))
\]

\[
= s(u \cos(\psi - \psi_p) - v \sin(\psi - \psi_p)) - \dot{\omega} \sqrt{\left(\frac{dx_p}{d\omega}\right)^2 + \left(\frac{dy_p}{d\omega}\right)^2} + e(u \sin(\psi - \psi_p) + v \cos(\psi - \psi_p)).
\]

Designing \( \psi \) and \( \dot{\omega} \) as follows [46]:

\[
\psi = \psi_c = \psi_p + \psi_r - \beta_d
= \psi_p + \arctan2(-e, k_c) - \arctan2(v, u)
\]
\[
\omega = \frac{u_d \cos(\psi - \psi_p) - v \sin(\psi - \psi_p) + k_s \varepsilon}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}},
\]

where \( k_e > 0 \) and \( k_s > 0 \). Substituting Equation (14) and Equation (15) into Equation (13) yields:

\[
V_e = -k_s \varepsilon^2 \frac{u \sqrt{u^2 + \varepsilon^2} \varepsilon^2}{|u| \sqrt{\varepsilon^2 + (k_e)^2}} e^2 = -k_s \varepsilon^2 \frac{\sqrt{u^2 + \varepsilon^2} \varepsilon^2}{\sqrt{\varepsilon^2 + (k_e)^2}} \leq 0 (i f u \geq 0).
\]

According to the derivation above, the desired yaw angle \( \psi \) and the derivative of the path parameter \( \dot{\psi} \) are obtained. Since the vertical plane motion is not considered, the desired attitude \( \Theta_e \) is also obtained.

3.2. MDP Model of the Airship

MDP is a discrete-time stochastic control process, providing a mathematical framework for optimization problems solved via RL. MDP can be expressed by a tuple of four elements \((S, A, P, R)\).

In time step \( t \), the state of the agent is \( S_t \), and the controller chooses an action \( A_t \) based on \( S_t \), independent of all previous states and actions. In next time step \( t + 1 \), the agent moves to state \( S_{t+1} \) and gives the controller an immediate reward \( R \), according to the state transition probability \( P \). The airship’s MDP model is described as follows.

First, state space \( S \). A qualifying state should contain sufficient information for the controller to make reasonable decisions, and avoid invalid information to accelerate the optimization process. In the path following control scenario, the state space \( S \) is selected as follows:

\[
S = \{s_1, s_2, s_3, \ldots, s_t, \ldots\}, s_t = (r, u, v, \Delta \psi, \Delta x, \Delta y, \Delta u, F_T, \delta_r),
\]

where \( \Delta \psi = \psi - \psi_c \), \( \Delta x = x - x_c \), and \( \Delta y = y - y_c \). Considering the inherent feature of airships and the guidance loop, we have a bounded state space, where \( r \in [-r_{max}, r_{max}], u \in [0, u_{max}], v \in [0, v_{max}], \Delta \psi \in [-\pi, \pi], \Delta x \in (-K_x, K_x), \Delta y \in (-K_y, K_y), \Delta u \in [-u_{max}, u_{max}], F_T \in [-F_{max}, F_{max}], \) and \( \delta_r \in [-\delta_{max}, \delta_{max}] \).

Second, action space \( A \). To deal with the acceleration and rate saturation problem, the acceleration of actuators is selected as the continuous action space, as well as the output of the PPO network. \( sat_{rate}(a) \) represents the acceleration saturation while \( sat_{magnitude}(F) \) denotes the control variable saturation.

\[
sat_{rate}(a_t) = \begin{cases} 
    a_{min}, a_t < a_{min} \\
    a_{min}, a_{min} \leq a_t \leq a_{max} \\
    a_{max}, a_t > a_{max}
\end{cases}
\]

\[
F_t = F_{t-1} + \int_{t-\Delta t}^{t} a dt
\]

\[
sat_{magnitude}(F_t) = \begin{cases} 
    F_{min}, F_t < F_{min} \\
    F_{t}, F_{min} \leq F_t \leq F_{max} \\
    F_{max}, F_t > F_{max}
\end{cases}
\]

Third, state transition probability \( P \). In this study, the state transition is determined by the airship model. In other words, the result of an action in a certain state is unique.

Fourth, reward function \( R \). The reward function is described as:

\[
\begin{align*}
    D_t &= \sqrt{\Delta x^2 + \Delta y^2} \\
    R_t &= k_D (D_t - D_{t-1}) - k_\psi |\Delta \psi| - k_u |\Delta u|
\end{align*}
\]

where \( k_D < 0, k_\psi > 0, \) and \( k_u > 0 \).
3.3. Optimization Process of PPO

Policy gradient methods generally consist of a policy gradient estimator and a stochastic gradient ascent algorithm. Usually, the estimator is obtained by differentiating an objective function whose gradient is designed as the policy gradient estimator. In PPO, the clipped objective function [40] is given by:

\[ L_t^{\text{CLIP}+\text{VF}+S} = \hat{E}_t \left[ L_t^{\text{CLIP}}(\theta) - c_1 L_t^{\text{VF}}(\theta) + c_2 S \right], \]  

(22)

where \( L_t^{\text{VF}}(\theta) = (V_{\theta}(s_t) - V_{\theta}(s_t))^{2} \), \( S \) denotes an entropy bonus, and \( L_t^{\text{CLIP}}(\theta) \) is defined as:

\[ L_t^{\text{CLIP}}(\theta) = \hat{E}_t \left[ \min (r_t(\theta) \delta_t A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \delta_t A_t) \right], \]

(23)

where \( r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \) is the probability ratio, \( \pi \) is the action policy, \( \theta \) is the policy parameter, and \( \epsilon \) is a clip hyperparameter. \( \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \) restricts aggressive policy updates, thus achieving a satisfying performance. \( \delta_t A_t \) is the advantage estimator given by:

\[ \delta_t A_t = \delta_t + (\gamma \lambda) \delta_{t+1} + \cdots + (\gamma \lambda)^{T-t+1} \delta_T, \]

(24)

where \( \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \), and \( T \) is a constant much less than the episode length.

The PPO (see Algorithm 1) adopted in this paper includes an actor network and a critic network. As shown in Figure 2, the input layer and hidden layer of the actor and the critic share the same structure. The input states pass through two full connection layers and finally connect to the output layer. The output layer of the actor is a full connection layer with one output, representing the state value.

Algorithm 1 PPO.

1: Input: initial actor-network parameter \( \theta_0 \) and critic-network parameter \( \phi_0 \).
2: for \( k = 1, 2, \ldots \) do
3: Run policy \( \pi_{\theta_k} \) in the environment for \( T \) timesteps.
4: Compute rewards-to-go \( R_t \).
5: Compute advantage estimates \( \hat{A}_t \) based on the current critic network \( V_{\phi_k} \).
6: Update the actor network by maximizing the clipped objective function Equation (22) via gradient ascent with Adam [47]:
7: \[ \theta_{k+1} = \arg \max_{\theta} L_t^{\text{CLIP}+\text{VF}+S}(\theta) \]
8: Update the critic network via gradient descent algorithm:
9: \[ \phi_{k+1} = \arg \max_{\phi} (V_{\phi} - R_t)^2 \]
10: end for

During training, the processed states and target of the airship are taken as the input of the neural network. The actor-network outputs the derivation of the airship input based on the processed states, while the critic network learns a value function as the actor update basis. The output of the neural network is the acceleration of the control input. After saturation and integration, the control torque and force are obtained.

4. Simulations

In this section, five sets of simulations are presented. In the first simulation, the controller is trained by following a circular arc path. To evaluate the performance of the well-trained controller, a comparative path following controller based on nonlinear MPC is introduced, and some simulations are implemented. Subsequently, the performance of the proposed controller is evaluated by following
the circular, general, and broken line paths without any adjustment in the next three simulations, respectively. The last simulation is conducted under wind disturbance.

4.1. Controller Training and Comparing

During training, the initial states of the airship are $\zeta_0 = [-30 \text{ m}, -1530 \text{ m}]^T$, $\psi_0 = 2\pi (\text{rand}() - 0.5) \text{rad}$, $r_0 = 0 \text{ rad/s}$, $u_0 = 5 \text{ m/s}$, $v_0 = 0 \text{ m/s}$. $\text{rand}()$ is a random function on the open interval $(0, 1)$ generating uniformly distributed pseudorandom numbers. The desired path is given as:

$$\zeta_p(\omega) = [1500 \sin(3.0\omega/1500 - \pi), 1500 \cos(3.0\omega/1500 - \pi)]^T \tag{25}$$

The physical parameters of the airship are the same as those in Reference [9]. Some of the parameters of the airship and the controller are listed in Table 1. After 15,000 episodes of training, the episode reward of the proposed method converges to 15 on average from 150.

### Table 1. Parameters of the airship and the controller.

| Parameters | Value | Parameters | Value |
|------------|-------|------------|-------|
| $\gamma$  | 0.995 | $\lambda$ | 0.97  |
| $T$        | 256   | $\varepsilon$ | 0.2 |
| $F_{\text{max}}$ | 150 | $F_{\text{min}}$ | 0 |
| $\delta_{r, \text{max}}$ | 10 | $\delta_{r, \text{min}}$ | -10 |
| $a_{F, \text{max}}$ | 100 | $a_{F, \text{min}}$ | -100 |
| $a_{\psi, \text{max}}$ | 45 | $a_{\psi, \text{min}}$ | -45 |
| $k_e$  | 30    | $k_s$ | 0.2 |
| $k_D$ | -0.1  | $k_\psi$ | 1.2 |
| $k_u$ | 0.25  |          |       |

As a comparison, an extended nonlinear MPC algorithm for time-varying reference in Reference [44] is introduced. The simulations are performed on an 8-core machine with an Intel Core i5-9300H CPU at 2.40 GHz and 16 GB of RAM. The detailed expression of the algorithm is presented in Algorithm 3.11, Chapter 3 in Reference [44].

To estimate the performance of the two controllers, we firstly perform four scenes with different initial states under the same task. The simulation time is set as 200 s, the time step is 0.1 s, and the simulation results are shown in Table 2 and Figure 3. It is obvious in Table 2 that the time consumption of the compared MPC method, averaging 206.63 s, is far greater than the proposed method. That’s even longer than the simulation time, which is unacceptable for practical application. In contrast, the proposed method takes only an average of 1.15 s, demonstrating a satisfying real-time performance in the path-following process.

### Table 2. Time consumption of the proposed and the compared methods.

| Time Consumption (s) | Scene 1 | Scene 2 | Scene 3 | Scene 4 | Average |
|----------------------|---------|---------|---------|---------|---------|
| proposed method      | 1.11    | 1.20    | 1.12    | 1.17    | 1.15    |
| compared method      | 213.14  | 199.15  | 167.92  | 246.31  | 206.63  |

As shown in Figure 3, the tracking trajectories and the yaw angles are illustrated. It must be emphasized that the desired yaw angles of the proposed and the compared controllers are inconsistent. The reason is that the desired yaw angles are related to the state variables, such as the position and velocity. Due to the differences between the algorithms, different control inputs are obtained, leading to different state variables as well as the desired yaw angles.
Figure 3. Tracking trajectories and time histories of yaw angles with different initial states for the proposed and the compared controllers.

For both methods, the trajectories and the yaw angles are convergent in all four scenes. Besides, the greater the initial state differences are, the slower the convergence speeds of the trajectory and the yaw angle get. The distinctions lie that the tracking errors after convergence of the proposed method are 10–30 m, while they are less than 1 m of the compared method. The yaw angle error of the proposed method is $10^{-1}$–$10^{-2}$ rad, while it is less than $10^{-3}$ rad of the compared method. Furthermore, the convergence speed of the proposed method is faster than the compared, which is embodied as the trajectories and yaw angles approaching the desired curves with less time in the illustrations. In a word, the proposed method explicates faster convergence speed but worse in tracking precision.

To verify if the magnitude and rate saturation is solved, the state variables of the two controllers in Scene 3 are shown in Figure 4. The simulation time is set as 200 s, and the time step is 0.1 s. Figure 4a shows the time histories of angular velocity and velocities, illustrating that the forward velocity $u$ can track its desired value well. Although no desired values for $r$ and $v$ are set, they are convergent and bounded. Figure 4b gives the time histories of control inputs $F_T$ and $\delta_r$ with respect to time. Due to actuator rate constraints, the control inputs avoid an initial large jump in References [4,12], which is less harsh to the actuators. In addition, as can be seen from Figure 4b,c, the saturation constraints are never violated. However, the oscillation of the control inputs of the proposed method is relatively serious. Because the execution of the action policies is related to a probability density function, the control inputs are not always continuous.
In extreme circumstances where the airship is desired to follow a path with big initial tracking errors, strong saturation is inevitable. To compare the performance of the two controllers in such situations, a numerical simulation is presented in Figure 5. The time is set as 2000 s, and the time step is 0.1 s. The initial states are chosen as $\Delta x_0 = \Delta y_0 = 50$ m and $\psi_0 = -1.5280$ rad. For both controllers, the convergence speeds are much slower than Scene 1–4, consuming more than 300 s. To adjust $\psi$, the rudder keeps saturated for almost 300 s. During the adjustment, the magnitude and rate constraints are never violated. Besides, compared to the MPC method, the convergence speed of the proposed method is faster while the control inputs and states are less stable.
Figure 5. Scene 5: Path following with strong saturation. Initial states: $\Delta x_0 = \Delta y_0 = 50$ m, $\psi_0 = -1.5280$ rad.

Conclusions can be drawn from Figures 3–5 that the airships can track the desired path subject to actuators’ magnitude and rate saturation under either of the two controllers. Compared with the MPC method, the proposed one is faster in convergence speed but worse in tracking precision, and stronger in the oscillation of the control inputs. Meanwhile, the proposed method has a satisfying real-time performance which the compared method does not. Despite the tracking errors which are acceptable for the airship, the proposed controller has satisfactory effectiveness and robustness in the presence of actuators’ magnitude and rate saturation.

4.2. Circular Path Following

In this section, to validate the ability to transfer to new tasks, circular paths with different radii varying from 1 km to 2 km are adopt to the proposed controller. The simulation results are shown in Figures 6–8. As shown in Figure 6, the airship is capable of following all circular paths in the presence of magnitude and rate saturation. The average tracking errors are roughly less than 20 m, which implies the task-independent feature of the proposed controller. From Figures 7a and 8a, we learn the real yaw angle is quickly convergent to the desired yaw angle and remain stable. Besides, from the cycle of the airship yaw angles, we learn that the airship takes about 800 s to fly one round with a radius of 1 km, and 2000 s with a radius of 2 km, which implies that the airship has a larger speed when flying around a smaller circle. From Figures 7b and 8b, we learn that both of the desired forward speed $u$ is 6 m/s, but the slide speed $v$ is different. The slide speed is adjusted bigger to achieve a smaller turning radius. Thus, when flying a circle with a smaller radius, the same forward speed indicates the bigger resultant velocity, leading to a shorter endurance. We also learn that the velocity is tracked well, and the states are stable during path following tasks. It is worth emphasizing that the controller is directly obtained from Section 4.1 without adjustments, which proves the effectiveness of the proposed controller.

4.3. General Path Following

This section presents the results of the airship following the general paths of the proposed controller. As shown in Figures 9 and 10, two referenced paths composed of four linear segments and four arc segments are investigated. As can be seen from Figures 9 and 10, the controller accomplishes the tracking tasks well when encountering more general paths. Position errors are reduced efficiently and quickly to an acceptable region. Despite the errors between the referenced and real paths, the tracking states are quite stable.
**Figure 6.** Trajectories of following the circle paths with different radii.

**Figure 7.** Airship states when $R = 1.0$ km.

**Figure 8.** Airship states when $R = 2.0$ km.
Figure 9. Trajectories of following the general paths with different arcs and line segments.

(a) Time histories of yaw angles following Path 1. (b) Time histories of airship yaw angles following Path 2.

Figure 10. Time histories of yaw angles of the general paths.

4.4. Broken-line path following

In this section, a simulation of the airship tracking a continuous unsmoothed path is performed. The initial states are chosen as $\Delta x_0 = \Delta y_0 = -50 \text{ m}$ and $\psi_0 = 1.1980 \text{ rad}$. The desired path $\zeta_p (\omega) = [x_p (\omega), y_p (\omega)]^T$ is a broken-line containing cusps, which is illustrated in Figure 11, and the parameter equation is given as:

$$
\begin{align*}
x_p (\omega) &= \begin{cases} 
3\omega, & 0 \leq \omega < 50 \\
5\omega - 100, & 50 \leq \omega < 100 \\
\omega - 300, & 100 \leq \omega \leq 200 
\end{cases} \\
y_p (\omega) &= \begin{cases} 
3\omega, & 0 \leq \omega < 50 \\
5\omega - 100, & 50 \leq \omega < 100 \\
\omega - 300, & 100 \leq \omega \leq 200 
\end{cases}
\end{align*}
$$

(26)

As shown in Figure 11, the controller can track the broken-line path well, despite a brief adjustment at the cusp of the path. The tracking error of the yaw angle is quickly convergent to an acceptable
region. Besides, Figure 12 indicates that the angular velocity and velocities are relatively stable during the tracking process.

![Figure 11. Trajectories of following the broken-line path.](image)

**Figure 12.** Airship states when following the broken-line path.

4.5. Anti-Wind Path Following

In this section, the constant wind disturbance is taken into account when following the desired path. The simulation results are integrated and shown in Figure 13. The initial states are chosen as $\Delta x_0 = \Delta y_0 = 200$ m and $\psi_0 = -1.4810$ rad. The wind speed is 5 m/s, as expressed by the light blue arrows in Figure 13a.

Figure 13a illustrates that the airship is capable of following the desired path subject to magnitude and rate saturation, as well as the wind disturbance. Compared with Figure 5a, the airship is forced to cruise for extra miles to converge to the desired path due to the wind disturbance. Figure 13b shows that the yaw angle is well tracked, despite that the airship takes more than 500 s to converge to the desired value.

The simulation results prove the robustness of the controller. Although trained without disturbance, the controller is capable of resisting a wind disturbance.
5. Discussion

The proposed RL-based path following controller for underactuated airship shows satisfying performance in numerical simulation. Not only the magnitude and rate saturation of actuators are handled, but the capability of transferring between various tasks without adaption is obtained. Discussions focusing on the two aspects are presented next.

For the target orientation control system, actuators saturation is an inevitable problem. Generally, the actuators saturation problem is handled by auxiliary systems and backstepping technique [28,30,48,49], smooth hyperbolic functions [27,50], model predictive control [9,19,51,52], or employing the generalized nonquadratic cost function [53–55] to guarantee the constrained control input. However, the consideration of system capability is always behind the consideration of achieving the control objective. In fact, without first considering the system capability, the designed controllers will be less meaningful in practical application. Unlike the controllers aforementioned, the proposed control strategy is based on building a capacity envelope of the airship. The action policies are generated within the envelope, thus avoiding the saturation. This approach takes full advantage of the adaptivity and robustness of neural networks, more simple to design, and proves to be a feasible solution to the magnitude and rate saturation problems.

On the other hand, transferring to new tasks is a common and challenging problem for RL in practical applications. For airships path following control, the difficulty of transferring to new tasks lies in various task requirements. By reasonably defining the bounded state space $S$ and action space $A$, the policies and tasks are decoupled, indicating that the task-independent policies could transfer to new tasks without adaption.

However, there are still some limitations of the proposed method. First, the oscillation of the control input signals is relatively serious, which indicates a high-frequency actuation of the actuators, and will limit the practical implementation of the controller from shortening the service life of the actuators, occupying too much communication bandwidth, or consuming too much energy. Thus the proposed method needs further improvement to reduce the oscillation in the future. Second, the tracking precision of our approach is relatively low. On one hand, the defect in tracking accuracy is caused by the imperfect design of the reward function, which is considerably related to the algorithm accuracy. On the other, the robustness of the RL algorithm is acquired at the expense of tracking precision. There is no specific measurement to counteract the effect of disturbances. Therefore, although the episode reward converges to 10% of the beginning, there’s still room for improvement.

In the future, the oscillation of the control inputs will be reduced. Besides, the tracking precision of our algorithm will be improved. To reach that point, the design of the reward function might
be optimized by applying inverse RL methods, and the disturbance will be further considered by employing active observation or compensation. Furthermore, the way of applying the RL strategy to more complex scenarios like spatial flight will be researched.

To sum up, it is a meaningful attempt to combine the policy search strategy with the path following controller. It can be applied to not only airships but also other dynamic systems like vessels, spacecraft, and so forth [19,32,35,48], indicating the broad prospects of the approach. Moreover, the controller can transfer to various tasks without adaptation, and resist wind disturbance without anti-disturbance training, demonstrating the research potential of the RL method applying to the flight control field.

6. Conclusions

In this paper, a real-time path following controller has been presented for airships with actuator magnitude and rate saturation. First of all, the MDP model of the airship path following control is established, and the error bounded LOS guidance is proposed to restrain the state space. Next, a PPO algorithm containing an actor network and a critic network is utilized to acquire the optimal action policy through trial and error. Since the policy is obtained from the action space, the magnitude and rate saturation will not happen. Finally, the numerical simulation results show that the controller can track circle paths with different radii, general paths, and unsmoothed curves, without any adjustment of parameters. Moreover, even trained without disturbance, the controller shows the satisfying performance when confronting wind disturbance. All these results validate the effectiveness and robustness of the proposed controller. The unique feature of this study lies that the proposed error bounded LOS guidance law provides a task-independent state space for PPO, therefore the trained action policy can transfer to new tasks without adaptation. In the future, the oscillation of the control inputs will be reduced, the tracking precision will be raised by employing anti-disturbance techniques or optimizing the reward function, and RL-based spatial path following control will be studied.

Author Contributions: Conceptualization, W.L. and X.G.; methodology, W.L.; software, H.G.; validation, H.G., J.Y. and J.O.; formal analysis, H.G.; investigation, J.O.; resources, X.G.; data curation, H.G.; writing—original draft preparation, H.G.; writing—review and editing, W.L.; visualization, J.Y.; supervision, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References
1. Jamison, L.; Sommer, G.S.; Porche, I.R., III. High-Altitude Airships for the Future Force Army; Technical Report; Rand Arroyo Center: Santa Monica, CA, USA, 2005.
2. Schaefer, I.; Kueke, R.; Lindstrand, P. Airships as unmanned platforms: Challenge and chance. In Proceedings of the 1st UAV Conference, Portsmouth, VA, USA, 20–23 May 2002; p. 3423.
3. Khoury, G.A. Airship Technology; Cambridge University Press: Cambridge, UK, 2012; Volume 10.
4. Zheng, Z.; Xie, L. Finite-time path following control for a stratospheric airship with input saturation and error constraint. Int. J. Control 2019, 92, 368–393. [CrossRef]
5. Manikandan, M.; Pant, R.S. Design optimization of a tri-lobed solar powered stratospheric airship. Aerosp. Sci. Technol. 2019, 91, 255–262. [CrossRef]
6. Wang, Y.; Zhou, W.; Luo, J.; Yan, H.; Pu, H.; Peng, Y. Reliable intelligent path following control for a robotic airship against sensor faults. IEEE/ASME Trans. Mechatron. 2019, 24, 2572–2582. [CrossRef]
7. Zhou, W.; Zhou, P.; Wei, Y.; Duan, D. Finite-time spatial path following control for a robotic underactuated airship. IET Intell. Transp. Syst. 2019, 14, 449–454. [CrossRef]
8. Stoica, A.M.; Mujdei, L. A mixed H2/H-infinity jump markovian design approach for the automatic flight control system of an unmanned airship. Appl. Mech. Mater. Trans. Tech. Publ. 2015, 811, 179–185. [CrossRef]
9. Zheng, Z.; Yan, K.; Yu, S.; Zhu, B.; Zhu, M. Path following control for a stratospheric airship with actuator saturation. Trans. Inst. Meas. Control 2017, 39, 987–999. [CrossRef]
10. Zheng, Z.; Zou, Y. Adaptive integral LOS path following for an unmanned airship with uncertainties based on robust RBFNN backstepping. *ISA Trans.* 2016, 65, 210–219. [CrossRef]
11. Zuo, Z.; Cheng, L.; Wang, X.; Sun, K. Three-Dimensional Path-Following Backstepping Control for an Underactuated Stratospheric Airship. *IEEE Trans. Aerosp. Electron. Syst.* 2019, 55, 1483–1497. [CrossRef]
12. Zheng, Z.; Guan, Z.; Ma, Y.; Zhu, B. Constrained path-following control for an airship with uncertainties. *Eng. Appl. Artif. Intell.* 2019, 85, 295–306. [CrossRef]
13. Drik, M.; Kocamaz, A.F.; Castillo, O. Global Path Planning and Path-Following for Wheeled Mobile Robot Using a Novel Control Structure Based on a Vision Sensor. *Int. J. Fuzzy Syst.* 2020, 22, 1880–1891. [CrossRef]
14. Ontiveros-Robles, E.; Melin, P.; Castillo, O. Comparative analysis of noise robustness of type 2 fuzzy logic controllers. *Kybernetika* 2018, 54, 175–201. [CrossRef]
15. Yu, S.; Xu, G.; Zhong, K.; Ye, S.; Zhu, W. Direct-adaptive fuzzy predictive control for path following of stratospheric airship. In Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 5658–5664.
16. Castillo, O.; Cervantes, L.; Soria, J.; Sanchez, M.; Castro, J. A Generalized Type-2 Fuzzy Granular Approach with Applications to Aerospace. *Inf. Sci.* 2016, 354, 165–177. [CrossRef]
17. Castillo, O.; Amador Angulo, G.; Castro, J.; Garcia-Valdez, M. A Comparative Study of Type-1 Fuzzy Logic Systems, Interval Type-2 Fuzzy Logic Systems and Generalized Type-2 Fuzzy Logic Systems in Control Problems. *Inf. Sci.* 2016, 354, 257–274. [CrossRef]
18. Cervantes, L.; Castillo, O. Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. *Inf. Sci.* 2015, 324. [CrossRef]
19. Liu, C.; Li, C.; Li, W. Computationally efficient MPC for path following of underactuated marine vessels using projection neural network. *Neural Comput. Appl.* 2020, 32, 7455–7464. [CrossRef]
20. Qian, L.; Liu, H.H.T. Path-Following Control of a Quadrotor UAV with a Cable-Suspended Payload Under Wind Disturbances. *IEEE Trans. Ind. Electron.* 2020, 67, 2021–2029. [CrossRef]
21. Chen, D.; Liu, G. Coordinated Path-Following Control for Multiple Autonomous Vehicles with Communication Time Delays. *IEEE Trans. Control Syst. Technol.* 2020, 28, 2005–2012. [CrossRef]
22. Wang, H.; Zhou, F.; Li, Q.; Song, W. Reliable adaptive H∞ path following control for autonomous ground vehicles in finite frequency domain. *J. Frankl. Inst.* 2020, 357, 9599–9613. [CrossRef]
23. Lim, Y.H.; Ahn, H.S. Stability analysis of linear systems under state and rate saturations. *Automatica* 2013, 49, 496–502. [CrossRef]
24. Zhao, Z.; Yang, W.; Shi, H. Semi-global containment control of discrete-time linear systems with actuator position and rate saturation. *Neurocomputing* 2019, 349, 173–182. [CrossRef]
25. He, W.; He, X.; Ge, S. Boundary output feedback control of a flexible string system with input saturation. *Nonlinear Dyn.* 2015, 80. [CrossRef]
26. Zhao, Z.; Liu, Y.; Luo, F. Output feedback boundary control of an axially moving system with input saturation constraint. *ISA Trans.* 2017, 68. [CrossRef] [PubMed]
27. Liu, Z.; Liu, J.; He, W. Partial differential equation boundary control of a flexible manipulator with input saturation. *Int. J. Syst. Sci.* 2016, 48, 1–10. [CrossRef]
28. Yuan, J.; Zhu, M.; Guo, X.; Lou, W. trajectory tracking control for a stratospheric airship subject to constraints and Unknown disturbances. *IEEE Access* 2020, 8, 31453–31470. [CrossRef]
29. Chen, T.; Zhu, M.; Zheng, Z. Asymmetric error-constrained path-following control of a stratospheric airship with disturbances and actuator saturation. *Mech. Syst. Signal Process.* 2019, 119, 501–522. [CrossRef]
30. Xing, X.; Liu, J.; Wang, L. Modeling and vibration control of aero two-blade propeller with input magnitude and rate saturations. *Aerosp. Sci. Technol.* 2019, 84, 412–430. [CrossRef]
31. Lim, Y.H.; Ahn, H.S. Decentralized control of nonlinear interconnected systems under both amplitude and rate saturations. *Automatica* 2013, 49, 2551–2555. [CrossRef]
32. Liu, Z.; Liu, J.; Wang, L. Disturbance observer based attitude control for flexible spacecraft with input magnitude and rate constraints. *Aerosp. Sci. Technol.* 2018, 72, 486–492. [CrossRef]
33. Ji, N.; Liu, Z.; Liu, J.; He, W. Vibration control for a nonlinear three-dimensional Euler-Bernoulli beam under input magnitude and rate constraints. *Nonlinear Dyn.* 2018, 91, 2551–2570. [CrossRef]
34. Lou, W.; Guo, X. Adaptive trajectory tracking control using reinforcement learning for quadrotor. *Int. J. Adv. Robot. Syst.* 2016, 13, 38. [CrossRef]
35. Broida, J.; Linares, R. Spacecraft rendezvous guidance in cluttered environments via reinforcement learning. In Proceedings of the 29th AAS/AIAA Space Flight Mechanics Meeting, Ka’anapali, HI, USA, 13–17 January 2019; pp. 1–15.

36. Carlucho, I.; De Paula, M.; Wang, S.; Petillot, Y.; Acosta, G.G. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robot. Auton. Syst.* 2018, 107, 71–86. [CrossRef]

37. Chen, G.; Peng, Y.; Zhang, M. An Adaptive Clipping Approach for Proximal Policy Optimization. *arXiv* 2018, arXiv:1804.06461.

38. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* 2013, 32, 1238–1274. [CrossRef]

39. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.I.; Abbeel, P. Trust Region Policy Optimization. *arXiv* 2015, arXiv:1502.05477.

40. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* 2017, arXiv:1707.06347.

41. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1861–1870.

42. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1582–1591.

43. Landolfi, N.C.; Thomas, G.; Ma, T. A Model-based Approach for Sample-efficient Multi-task Reinforcement Learning. *arXiv* 2019, arXiv:1907.04964.

44. Grne, L.; Pannek, J. *Nonlinear Model Predictive Control*; Springer: London, UK, 2011.

45. Zheng, Z.; Huo, W. Planar path following control for stratospheric airship. *IET Control Theory Appl.* 2013, 7, 185–201. [CrossRef]

46. Zheng, Z.; Huo, W.; Wu, Z. Autonomous airship path following control: Theory and experiments. *Control Eng. Pract.* 2013, 21, 769–788. [CrossRef]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.