remote sensing

MDPI

Article

TD3-Based Optimization Framework for RSMA-Enhanced UAV-Aided Downlink Communications in Remote Areas

Tri-Hai Nguyen [1], Luong Vuong Nguyen [2], L. Minh Dang [3,4], Vinh Truong Hoang [5] and Laihyuk Park [1,*]

[1] Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; haint93@seoultech.ac.kr
[2] Department of Artificial Intelligence, FPT University, Da Nang 550000, Vietnam; vuongnl3@fe.edu.vn
[3] Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam; danglienminh@duytan.edu.vn
[4] Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam
[5] Faculty of Computer Science, Ho Chi Minh City Open University, Ho Chi Minh City 700000, Vietnam; vinh.th@ou.edu.vn
[*] Correspondence: lhpark@seoultech.ac.kr

Abstract: The need for reliable wireless communication in remote areas has led to the adoption of unmanned aerial vehicles (UAVs) as flying base stations (FlyBSs). FlyBSs hover over a designated area to ensure continuous communication coverage for mobile users on the ground. Moreover, rate-splitting multiple access (RSMA) has emerged as a promising interference management scheme in multi-user communication systems. In this paper, we investigate an RSMA-enhanced FlyBS downlink communication system and formulate an optimization problem to maximize the sum-rate of users, taking into account the three-dimensional FlyBS trajectory and RSMA parameters. To address this continuous non-convex optimization problem, we propose a TD3-RFBS optimization framework based on the twin-delayed deep deterministic policy gradient (TD3). This framework overcomes the limitations associated with the overestimation issue encountered in the deep deterministic policy gradient (DDPG), a well-known deep reinforcement learning method. Our simulation results demonstrate that TD3-RFBS outperforms existing solutions for FlyBS downlink communication systems, indicating its potential as a solution for future wireless networks.

Keywords: downlink communication; flying base station; rate-splitting multiple access; twin-delayed deep deterministic policy gradient; unmanned aerial vehicle

check for updates

Citation: Nguyen, T.-H.; Nguyen, L.V.; Dang, L.M.; Hoang, V.T.; Park, L. TD3-Based Optimization Framework for RSMA-Enhanced UAV-Aided Downlink Communications in Remote Areas. *Remote Sens.* **2023**, *15*, 5284. https://doi.org/10.3390/rs15225284

Academic Editor: Francesco Nex

Received: 26 September 2023
Revised: 30 October 2023
Accepted: 7 November 2023
Published: 8 November 2023

1. Introduction

In recent years, the increasing demand for wireless communication networks has been driven by the widespread use of mobile devices, Internet of Things (IoT) applications, and smart cities. To meet this demand, the next generation of wireless technology (6G) is being developed to be even faster, more efficient, and more capable than its predecessors [1–4]. This advanced technology could enable innovative applications and services that were previously unthinkable. However, IoT devices can be installed in remote and isolated locations such as rural areas, mountains, and deserts. Establishing direct communication between IoT devices can be challenging in these areas due to their considerable distance. Additionally, ground base stations (BSs) are often absent in these remote areas due to the high economic costs. To address these challenges, aerial access networks and space–air–ground integrated networks have been developed [1–5]. Notably, due to exceptional merits such as high mobility, maneuverability, and flexibility, unmanned aerial vehicles (UAVs) can be deployed as flying base stations (FlyBSs or UAV-BSs), which enables rapid deployment in remote locations, disaster-stricken zones, or temporary events [4,5]. In addition, UAVs can also serve as wireless power transmitters, enabling energy delivery to ground users via wireless power transfer technology [6].

Concurrently, as the number of mobile users continues to grow, the available spectrum resources have diminished significantly. In light of this, multiple access techniques have become increasingly essential [7]. Recently, rate-splitting multiple access (RSMA) has been proposed as a novel multiple access scheme to improve the efficiency of multi-user communication systems [8,9]. Previous research has shown that RSMA outperforms other advanced multiple access techniques, i.e., non-orthogonal multiple access (NOMA) [10,11]. Specifically, NOMA requires a single user to decode the messages of other co-scheduled users to obtain its intended message [7], which reduces the communication performance. In contrast, RSMA divides each user's message into two parts: common and private. All users can decode the common part, while the intended recipients only decode the private parts. Then, the original message can be reconstructed from the common part and the private part via a successive interference cancellation (SIC) technique [12]. RSMA can be used with multi-antenna transmission to achieve an optimal performance with high coverage. Several research efforts have been devoted to RSMA [10,11,13–18]. However, they have been limited to terrestrial BSs. On the other hand, FlyBSs offer greater freedom in system design, enabling new and innovative applications. For example, high-altitude UAVs with RSMA have assisted in computation offloading from IoT devices and smart vehicles [19,20]. However, these platforms are typically stationary in one stratosphere location, limiting their mobility and flexibility. Additionally, they can be more expensive to build and maintain than low-altitude UAVs and require a longer deployment time. Prior work on RSMA-based FlyBS downlink communication systems has focused on developing optimization techniques to improve the performance, but none of these studies have considered time-varying environments [21–24]. In [25], multiple UAVs assist a BS in providing RSMA communication services to ground users; however, they are deployed in fixed locations. In [26], a FlyBS downlink communication system with RSMA is proposed; however, it is limited to two-dimensional (2D) space, which restricts the system's performance. To fully exploit the FlyBS's high mobility and flexibility, it should be operated in three-dimensional (3D) space. In addition, optimizing RSMA-enabled FlyBS communications poses challenges due to the complex interactions between the parameters of RSMA, FlyBS, and mobile users. Traditional optimization methods may not solve these problems efficiently, especially in dynamic environments where the channel conditions and user mobility change frequently. Therefore, the optimization aspect of RSMA-enabled FlyBS systems with 3D trajectory continues to be an open area of research.

Recent advances in machine learning have led to the emergence of deep reinforcement learning (DRL) [27], a powerful technique that combines reinforcement learning and deep learning to solve a wide range of optimization problems [28–32]. Deep Q-network (DQN) is the first DRL algorithm [27]. DQN approximates the Q-function using a deep neural network (DNN), which estimates the expected cumulative reward for taking a specific action in a particular state and following a certain policy. The Q-function is then used to select the action with the highest expected reward. However, DQN can be unstable to train and slow to learn in complex environments. Additionally, when using DQN to solve optimization problems, the action space must be discretized, which can reduce the optimization performance. In contrast to DQN, the deep deterministic policy gradient (DDPG) is specifically designed for scenarios involving high-dimensional continuous action spaces [33]. DDPG employs an actor–critic architecture consisting of two DNNs: an actor network and a critic network. The actor network is responsible for taking the optimal action in a given state, while the critic network evaluates the quality of action chosen by the actor network. At present, many studies have applied the DDPG algorithm to wireless communication systems [19,20,26,34,35]. However, one shortcoming of DDPG is that the learned Q-function often overestimates the Q-values, resulting in significant errors in the policy. To address this issue, the twin-delayed deep deterministic policy gradient (TD3) algorithm, a more recent DRL approach, has been introduced [36]. It uses three critical modifications to DDPG: clipped double Q-learning, target policy smoothing, and delayed policy updates. TD3 is more stable and robust than DDPG and it can perform better than other DRL techniques in various tasks [36]. However, to the best of our knowledge, there is no existing research on applying TD3 to optimize RSMA-enabled FlyBS systems with 3D trajectories.

Motivated by the above discussion, this paper proposes a TD3-based optimization framework, TD3-RFBS, to optimize an RSMA-enhanced FlyBS system with a 3D trajectory. The system has potential practical implications, such as facilitating emergency communication services in disaster zones, connecting rural communities to the Internet, and enabling remote monitoring and control of critical infrastructure. The key contributions of this work can be summarized as follows:

- We introduce an RSMA-enhanced FlyBS system, where the FlyBS equipped with a multi-antenna array serves mobile ground users in hard-to-reach areas. At the same time, the communication channel is improved by RSMA technology. To maximize the downlink sum rate, we formulate an optimization problem that considers the 3D FlyBS trajectory and RSMA parameters, i.e., the precoding matrix and common rate vector, while considering the mobility of ground users.
- We transform the problem into a Markov decision process (MDP) by carefully defining the state space, action space, and reward function. To solve the MDP model, we develop the TD3-RFBS optimization framework, which stands for Twin-Delayed Deep Deterministic policy gradient for Rate-splitting multiple access-enhanced Flying Base Station. The TD3 algorithm is used to overcome the overestimation bias issue present in the well-known DDPG algorithm. In the framework, the FlyBS engages, monitors, and acquires knowledge of channel patterns without any pre-existing channel state information (CSI) to optimize its actions.
- We conduct extensive simulations to evaluate the performance of the TD3-RFBS framework. The results confirm that the framework outperforms baseline solutions, including DDPG and local search-based counterparts regarding the learning convergence and total achievable rate.

Structure: The rest of this paper is organized as follows. Section 2 reviews the background and related work. Section 3 presents the system model and problem formulation. Section 4 describes the MDP model and introduces our proposed framework based on the TD3 algorithm. The simulation results and performance analysis are presented in Section 5. Finally, we conclude the paper in Section 6.

Notations: Matrices and vectors are represented by boldface uppercase and lowercase symbols, respectively. The transpose, Hermitian transpose, and trace of a matrix are denoted by $(\cdot)^{\mathrm{T}}$, $(\cdot)^{\mathrm{H}}$, and $\mathrm{tr}(\cdot)$, respectively. A complex number's real and imaginary parts are represented by $\Re(\cdot)$ and $\Im(\cdot)$. $\mathbb{E}(\cdot)$ denotes the expectation operator, $\otimes$ denotes the Kronecker product, $||\cdot||$ denotes the Euclidean norm, $|\cdot|$ denotes the absolute value, and $\mathbf{I}$ denotes the identity matrix.

2. Related Work

Recently, rate-splitting multiple access (RSMA) technology has emerged as a promising approach for advancing next-generation mobile networks [8,9]. Table 1 presents a comprehensive comparison of studies closely related to our work on downlink RSMA-based communication systems. According to the principles of RSMA, each message intended for a user is partitioned into two parts: a common part and a private part. The common parts are merged into a single common signal, and the private part is encoded into a private signal for each user individually. By decoding interference partially and treating the remainder as noise, RSMA enables effective interference management. This allows RSMA to enhance various aspects of communication systems, such as reliability, energy efficiency, spectrum efficiency, and quality of service (QoS). RSMA outperforms other multiple access techniques by offering greater flexibility and powerful interference management capabilities [10]. In downlink RSMA, numerous studies have focused on optimizing the transmission power and precoding vectors for both the common and private signals to achieve various objectives, such as sum-rate maximization [10,13–15], max–min rate fairness [16,17], and energy efficiency maximization [11,18].

Table 1. A comparison with existing studies on downlink RSMA-based communication systems.

References	Optimization Objective	Optimization Method	FlyBS	FlyBS Trajectory	Time-Varying Environment
[10]	Sum-rate maximization	Alternating optimization	✗	Not applicable	✗
[11]	Energy efficiency	Successive convex approximation	✗	Not applicable	✗
[13]	Sum-rate maximization	DRL (i.e., PPO)	✗	Not applicable	✓
[14]	Sum-rate maximization	Successive convex approximation	✗	Not applicable	✗
[15]	Sum-rate maximization	Evolutionary game	✗	Not applicable	✓
[16]	Max–min rate fairness	Iterative algorithm	✗	Not applicable	✗
[17]	Max–min rate fairness	Successive convex approximation	✗	Not applicable	✗
[18]	Energy efficiency	Successive convex approximation	✗	Not applicable	✗
[21]	Sum-rate maximization	Alternating optimization	✓	Optimal position	✗
[22]	Sum-rate maximization	Alternating optimization	✓	Optimal position	✗
[23]	Sum-rate maximization	Alternating optimization	✓	Optimal position	✗
[24]	Energy efficiency	Sub-problem decomposition	✓	Optimal position	✗
[25]	Sum-rate maximization	DRL (i.e., PPO)	✓	Fixed position	✓
[26]	Sum-rate maximization	DRL (i.e., DDPG)	✓	2D trajectory	✓
Our work	Sum-rate maximization	DRL (i.e., TD3)	✓	3D trajectory	✓

The symbol ✓ is used to denote that an aspect is included in the study, while ✗ indicates that it is not.

Nonetheless, the studies above have predominantly focused on terrestrial-fixed BSs. In contrast, UAV-aided communication systems offer greater cost-effectiveness and potential for improved QoS due to their high mobility, on-demand coverage, and ability to establish line-of-sight (LoS) links compared to their terrestrial counterparts. In [21], the authors explored the simultaneous optimization of the UAV position and RSMA variables to maximize the downlink weighted sum-rate in a UAV-aided communication system. However, their objective was to find the optimal placement for UAV deployment, not the UAV trajectory. Similarly, several studies [22–24] investigated the performance of downlink communication in a UAV-BS setting utilizing RSMA to serve multiple users. These works adopted alternating optimization, sub-problem decomposition, and iterative approaches, which are problem-specific, challenging to extend to general cases, and do not consider the time-varying environment [21–24]. On another front, DRL has been widely used to solve non-convex optimization problems in wireless communication systems [13,19,20,25,26,34,35,37]. The work [25] proposed a multi-UAV-assisted BS system to deliver communication services to ground users with both downlink and uplink RSMA transmissions. It used proximal policy optimization (PPO) to solve the sum-rate maximization problem in the system. However, PPO can be slow to converge, especially in complex environments, and it requires a large amount of data to train effectively, which can be a challenge in some applications. In [34], the authors utilized a UAV with an intelligent reflecting surface as a relay to assist downlink communications from a terrestrial BS to ground users. In [19,20,37], RSMA-based high-altitude UAVs assisted IoT devices or connected vehicles with computation offloading. In [26], the authors investigated a UAV-supported downlink communication system with RSMA. However, these studies limit the UAV flying trajectory to a 2D space or deploy the UAVs in a fixed position, which restricts the mobility and flexibility of the UAVs [19,20,25,26]. In addition, the well-known DDPG algorithm, which is used in [19,20,26,34,37], suffers from the overestimation problem, which can degrade the performance of wireless communication systems. The TD3 algorithm can address this issue through three improved techniques: clipped double Q-learning, target policy smoothing, and delayed policy updates. TD3 has proven more efficient than DDPG, PPO, and other DRL approaches in various learning tasks [36].

Different from the previous studies, we focus on a multi-antenna FlyBS with RSMA that can fly along a 3D trajectory while providing communication services to mobile ground users. The multi-antenna technology has been extensively employed in ground-based BSs [10,11,13–18]. By leveraging the benefits of increased signal strength, expanded coverage, reduced interference, and beamforming capabilities, multi-antenna arrays can provide reliable and high-quality wireless services. With recent technological advancements such as miniaturized antennas, low-power electronics, and improved signal processing techniques, the integration of multi-antenna arrays into FlyBSs holds immense potential for enhancing communication capabilities in remote and underserved areas [21,24–26]. Lastly, we leverage the TD3 algorithm to solve the optimization problem rather than the well-known DDPG algorithm, which suffers from an overestimation bias.

3. System Model and Problem Formulation

This section begins with an introduction to the system model for a FlyBS system enhanced by RSMA, including mobility, channel, and signal models. Following this, the problem of maximizing the sum-rate is formulated.

3.1. System Model

We investigate the RSMA-enhanced FlyBS model depicted in Figure 1. The system comprises a UAV-BS or FlyBS equipped with a uniform rectangular array containing $N = N_1 \times N_2$ antennas and K single-antenna ground users (GUs or sensors devices installed in remote areas). Due to obstacles such as high buildings, mountains, or long distances, a direct link between the ground BS and the GUs is unavailable. Therefore, the FlyBS is used to transmit data to GUs. The group of K GUs is denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$. Similar to [4,20,34], the operation time is divided into T equal time slots, each with a duration of τ, which is sufficiently small to assume that the network topology remains constant within each time slot. The collection of time slots is represented as $\mathcal{T} = \{1, 2, \ldots, T\}$.

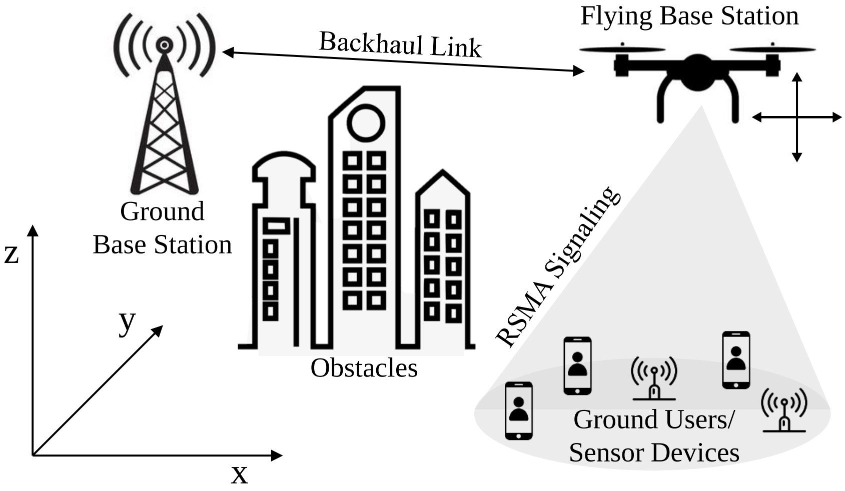

Figure 1. RSMA-enhanced FlyBS system.

3.1.1. Mobility Model

The location of the FlyBS and GU k are denoted by $\mathbf{q}_0(t) = (x_0(t), y_0(t), z_0(t))$ and $\mathbf{q}_k(t) = (x_k(t), y_k(t), z_k(t))$, respectively, in a 3D Cartesian coordinate system. Accordingly, the distance between the FlyBS and GU k can be calculated by

$$d_k(t) = \sqrt{(x_0(t) - x_k(t))^2 + (y_0(t) - y_k(t))^2 + (z_0(t) - z_k(t))^2}. \tag{1}$$

During time slot t with duration τ, the FlyBS can fly towards an azimuth angle of $\phi_0(t) \in [0, 2\pi]$ and an elevation angle of $\theta_0(t) \in [0, \pi]$ with a velocity of $v_0(t) \in [0, v_{0,\max}]$ (m/s), where $v_{0,\max}$ denotes the maximum velocity of FlyBS. Hence, its mobility at time slot $t+1$ can be expressed by [34]

$$\begin{cases} x_0(t+1) = x_0(t) + v_0(t)\tau \sin(\theta_0(t)) \cos(\phi_0(t)) \\ y_0(t+1) = y_0(t) + v_0(t)\tau \sin(\theta_0(t)) \sin(\phi_0(t)) \\ z_0(t+1) = z_0(t) + v_0(t)\tau \cos(\theta_0(t)) \end{cases}. \tag{2}$$

We assume that each GU k moves randomly within the considered area with a velocity $v_k(t) \in [0, v_{k,\max}]$ (m/s), where $v_{k,\max}$ denotes the maximum velocity of the GU k. In addition, the operation of FlyBS is limited in a region of $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ with a height between $[z_{\min}, z_{\max}]$. Thus, we have

$$\begin{cases} x_{\min} \leq x_0(t) \leq x_{\max} \\ y_{\min} \leq y_0(t) \leq y_{\max} \\ z_{\min} \leq z_0(t) \leq z_{\max} \end{cases}. \tag{3}$$

3.1.2. Channel Model

In multi-antenna technology, a uniform rectangular array (URA) offers the advantage of higher gain while maintaining a compact size compared to a traditional uniform linear array [38]. By employing radiation beam patterns in both elevation and azimuth planes, URAs provide additional degrees of freedom that enhance interference suppression, wireless network coverage, and system capacity. Therefore, we consider equipping the FlyBS with the URA of dimensions $N = N_1 \times N_2$, as illustrated in Figure 2.

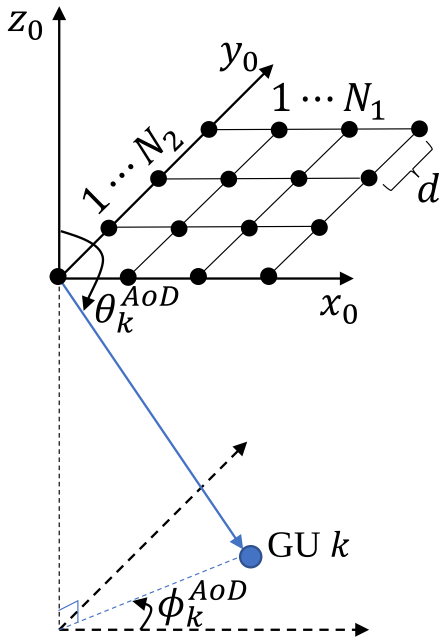

Figure 2. Geometry of URA with $N = N_1 \times N_2$ antennas.

For simplicity of presentation, we omit the time slot index in the following discussion. Let $\mathbf{h}_k \in \mathbb{C}^{N\times 1}$ represent the air-to-ground channel from the FlyBS to GU k. The channel $\mathbf{h}_k$ is assumed to follow a widely recognized Rician fading model [26,34,35]. Consequently, the channel connecting the FlyBs and GU k can be expressed as

$$\mathbf{h}_k = L_k\left(\sqrt{\frac{\kappa}{\kappa+1}}\mathbf{h}_k^{LoS} + \sqrt{\frac{1}{\kappa+1}}\mathbf{h}_k^{NLoS}\right), \tag{4}$$

where $\mathbf{h}_k^{LoS}$ is the deterministic LoS component, $\mathbf{h}_k^{NLoS} \sim \mathcal{CN}(0,1)$ is the non-line-of-sight (NLoS) component, κ is the Rician factor, and L_k is the distance-dependent large-scale path loss. According to [26,34,38], since the FlyBS equipped the URA, the LoS component of the channel $\mathbf{h}_k^{LoS}$ is computed by

$$\mathbf{h}_k^{LoS} = \mathbf{a}_k\left(\phi_k^{AoD}, \theta_k^{AoD}\right) = \mathbf{a}_{k,N_1}\left(\phi_k^{AoD}, \theta_k^{AoD}\right) \otimes \mathbf{a}_{k,N_2}\left(\phi_k^{AoD}, \theta_k^{AoD}\right), \tag{5}$$

where ϕ_k^{AoD} and θ_k^{AoD} represent the azimuth and elevation angle-of-departure (AoD) at the FlyBS to GU k, respectively, and $\mathbf{a}_k \in \mathbb{C}^{N\times 1}$ represents the antenna array response between the FlyBS and GU k. Furthermore, $\mathbf{a}_{k,N_1}\left(\phi_k^{AoD}, \theta_k^{AoD}\right)$ and $\mathbf{a}_{k,N_2}\left(\phi_k^{AoD}, \theta_k^{AoD}\right)$ can be calculated as

$$\begin{aligned}
\mathbf{a}_{k,N_1}\left(\phi_k^{AoD}, \theta_k^{AoD}\right) &= \left[1, e^{j\frac{2\pi}{\lambda}d\cos\phi_k^{AoD}\sin\theta_k^{AoD}}, \ldots, e^{j\frac{2\pi}{\lambda}(N_1-1)d\cos\phi_k^{AoD}\sin\theta_k^{AoD}}\right]^{\mathrm{T}}, \\
\mathbf{a}_{k,N_2}\left(\phi_k^{AoD}, \theta_k^{AoD}\right) &= \left[1, e^{j\frac{2\pi}{\lambda}d\sin\phi_k^{AoD}\sin\theta_k^{AoD}}, \ldots, e^{j\frac{2\pi}{\lambda}(N_2-1)d\sin\phi_k^{AoD}\sin\theta_k^{AoD}}\right]^{\mathrm{T}},
\end{aligned} \tag{6}$$

where λ represents the carrier wavelength, d represents the distance between the antennas, $\cos\phi_k^{AoD}\sin\theta_k^{AoD} = \frac{x_0-x_k}{d_k}$, and $\sin\phi_k^{AoD}\sin\theta_k^{AoD} = \frac{y_0-y_k}{d_k}$.

3.1.3. Signal Model

Following the one-layer RSMA principle [8,10], the FlyBS divides the intended message for each GU k into common and private parts. Next, the common part messages of all GUs are encoded using a shared codebook into a common signal s_0, designed to reduce interference. All GUs subsequently decode the common signal s_0. On the other hand, the private part of the message for each GU is encoded using an independent codebook into a dedicated private signal $s_k, \forall k \in \mathcal{K}$, which can only be decoded by the corresponding GU. It is important to emphasize that the shared codebook for the common signal is accessible to all GUs, while the codebooks for the private signals are exclusively known by their corresponding GUs. This distinction enables each GU to differentiate between its private signal and the private signals of other GUs [8]. Accordingly, the vector comprising the $K+1$ signals for transmission is represented as $\mathbf{s} = [s_0, s_1, \ldots, s_K]^{\mathrm{T}}$, with $\mathbb{E}(\mathbf{s}\mathbf{s}^{\mathrm{H}}) = \mathbf{I}$. A precoding matrix $\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_K] \in \mathbb{C}^{N \times (K+1)}$ is used to precode the signals. Here, $\mathbf{p}_i \in \mathbb{C}^{N \times 1}$ is the linear precoder corresponding to the signal $s_i, \forall i \in \{0, 1, \ldots, K\}$. Thus, the received signal r_k at GU k can be expressed as

$$\begin{aligned} r_k &= \mathbf{h}_k^{\mathrm{H}} \mathbf{P} \mathbf{s} + n_k \\ &= \underbrace{\mathbf{h}_k^{\mathrm{H}} \mathbf{p}_0 s_0}_{\text{desired common signal}} + \underbrace{\mathbf{h}_k^{\mathrm{H}} \mathbf{p}_k s_k}_{\text{desired private signal}} + \underbrace{\sum_{i \in \mathcal{K} \setminus \{k\}} \mathbf{h}_k^{\mathrm{H}} \mathbf{p}_i s_i}_{\text{interference}} + \underbrace{n_k}_{\text{noise}}, \end{aligned} \tag{7}$$

where $n_k \sim \mathcal{CN}(0, \sigma^2)$ represents the additive white Gaussian noise at GU k with noise power σ^2. The transmission power of FlyBS is limited by $\mathrm{tr}(\mathbf{P}\mathbf{P}^{\mathrm{H}}) \leq P_{\max}$, where $P_{\max}$ represents the maximum transmit power [11,21].

The decoding process at GU k is as follows. Initially, GU k decodes the common signal s_0 by considering all private signals as noise. Once s_0 is completely decoded, it is eliminated from the obtained signals using the SIC technique, enabling the extraction of private signals [12]. Afterward, GU k decodes its desired private signal s_k by considering the private signals of other GUs as noise. By combining the common part and the private part from the decoded signals, GU k can retrieve its message. The signal-to-interference-plus-noise ratio (SINR) for the common signal s_0 and private signal s_k at GU k can be respectively determined by

$$\begin{aligned} \gamma_k^c &= \frac{\left|\mathbf{h}_k^{\mathrm{H}} \mathbf{p}_0\right|^2}{\sum_{i \in \mathcal{K}} \left|\mathbf{h}_i^{\mathrm{H}} \mathbf{p}_i\right|^2 + \sigma^2}, \\ \gamma_k^p &= \frac{\left|\mathbf{h}_k^{\mathrm{H}} \mathbf{p}_k\right|^2}{\sum_{i \in \mathcal{K} \setminus \{k\}} \left|\mathbf{h}_i^{\mathrm{H}} \mathbf{p}_i\right|^2 + \sigma^2}. \end{aligned} \tag{8}$$

Accordingly, the achievable rate of decoding s_0 and s_k at GU k is denoted as $R_k^j = B \log_2(1 + \gamma_k^j), \forall j \in \{c, p\}$, where B is the bandwidth. Furthermore, the attainable rate of the common signal is defined as $R_c = \min_{k \in \mathcal{K}} \{R_k^c\}$ to guarantee the successful decoding of the common signal s_0 by all GUs. Let C_k denote the part of the common rate allocated to GU k, such that $\sum_{k \in \mathcal{K}} C_k = R_c$. Thus, the overall achievable rate (in bits/second/Hertz, or bit/s/Hz) of GU k can be given as

$$R_k = C_k + R_k^p. \tag{9}$$

3.2. Problem Formulation

This paper focuses on the joint optimization of the precoding matrix $\mathbf{P}$, the common rate vector $\mathbf{c} = [C_1, C_2, \ldots, C_K]$, and the 3D trajectory of the FlyBS (taking into account the azimuth angle ϕ_0, elevation angle θ_0, and velocity v_0) to maximize the sum-rate of all GUs across all time slots. The optimization problem can be formulated as

$$\max_{\mathbf{P},\mathbf{c},\phi_0,\theta_0,v_0} \sum_{k\in\mathcal{K}} R_k \tag{10a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} C_k \leq R_c, \tag{10b}$$

$$C_k \geq 0, \forall k \in \mathcal{K}, \tag{10c}$$

$$R_k \geq R_{\min}, \forall k \in \mathcal{K}, \tag{10d}$$

$$\mathrm{tr}(\mathbf{P}\mathbf{P}^{\mathrm{H}}) \leq P_{\max}, \tag{10e}$$

$$0 \leq \phi_0 \leq 2\pi, 0 \leq \theta_0 \leq \pi, 0 \leq v_0 \leq v_{0,\max}, \tag{10f}$$

$$x_{\min} \leq x_0 \leq x_{\max}, y_{\min} \leq y_0 \leq y_{\max}, z_{\min} \leq z_0 \leq z_{\max}, \tag{10g}$$

where (10b) ensures that every GU can successfully decode the common signal, (10c) ensures that each GU's portion of the common rate is a positive value, (10d) guarantees the QoS for the GUs by a minimum required rate $R_{\min}$, and (10e), (10f), and (10g) ensure that the FlyBS's parameters, such as the transmit power, directional angles, velocity, and location, are within the accessible ranges. It can be observed that the defined problem is non-convex and difficult to be addressed by conventional optimization approaches. DRL shows its advantages in solving problems in highly dynamic environments [19,20,26,34,35]. The next section presents a DRL-inspired optimization framework to address the optimization problem.

4. TD3-RFBS Optimization Framework for RSMA-Enhanced FlyBS System

To tackle the challenges posed by the high complexity and dynamic nature of the optimization problem, we reformulate it as an MDP model. Subsequently, we propose a DRL-based algorithm under the TD3 framework to solve the MDP model.

4.1. Markov Decision Process Transformation

An MDP is a mathematical framework for modeling sequential decision-making problems. We transform the original problem into an MDP model, expressed as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma \rangle$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{R}$ denotes the reward function, and $\gamma \in [0, 1)$ denotes the discount factor. In this setup, the FlyBS acts as the agent, while the entire communication system is considered the environment. At each time step t, the agent observes its present state $s(t) \in \mathcal{S}$. Based on this observation, the agent selects an action $a(t) \in \mathcal{A}$. After executing the action $a(t)$, the agent moves to a new state $s(t+1)$ and receives an immediate reward $r(t)$. The agent aims to discover an optimum policy that maximizes the total reward, considering the discount factor γ. The discount factor determines how much the agent values future rewards relative to immediate rewards. When γ is set to 0, the agent only cares about immediate rewards. As γ increases, the agent places more weight on future rewards. Each component of the MDP model is explained in more detail below.

- State: At every time interval, the state $s(t)$ comprises the present location information of both the FlyBS and K GUs. By utilizing these observed locations, it is possible to estimate the CSI between the FlyBS and the respective GUs [34]. The state $s(t)$ is represented as

$$s(t) = \{\mathbf{q}_0(t), \mathbf{q}_k(t)\}, \forall k \in \mathcal{K}. \tag{11}$$

- Action: At each state $s(t)$, the agent makes decisions on the joint action $a(t)$, which encompasses the optimization variables of the precoding matrix $\mathbf{P}(t)$, the common rate vector $\mathbf{c}(t)$, and the parameters of the FlyBS (i.e., the azimuth angle $\phi_0(t)$, elevation angle $\theta_0(t)$, and velocity $v_0(t)$). It is formally given as

$$a(t) = \{\mathbf{P}(t), \mathbf{c}(t), \phi_0(t), \theta_0(t), v_0(t)\}, \tag{12}$$

where $\mathbf{p}_i(t) \in \mathbf{P}(t), \forall i \in \{0, 1, \ldots, K\}$ is the complex-valued linear precoder for the signal s_i and $p_{i,n}(t) \in \mathbf{p}_i(t) = \Re(p_{i,n}(t)) + j\Im(p_{i,n}(t)), \forall n \in \{1, 2, \ldots, N\}$. It is worth noting that the action defined in the above equation contains both discrete, continuous, and complex-valued variables which are not directly accessible to the DRL-based learning algorithms. To address this issue, we redefine the precoding matrix $\mathbf{P}(t)$ as $\overline{\mathbf{P}}(t) = [\overline{\mathbf{p}}_0(t), \overline{\mathbf{p}}_1(t), \dots, \overline{\mathbf{p}}_K(t)]$, where $\mathbf{p}_i(t) \in \mathbb{C}^{N\times 1}$ is reformed as $\overline{\mathbf{p}}_i(t) \in \mathbb{R}^{2N\times 1} = [\overline{p}_{i,1}(t), \overline{p}_{i,2}(t), \dots, \overline{p}_{i,2N}(t)], \forall i \in \{0, 1, \dots, K\}$ and $\overline{p}_{i,n}(t) \in [0,1], \forall n \in \{1, 2, \dots, 2N\}$ [35]. Hence, the original value of $p_{i,n}(t)$ can be computed by

$$\begin{aligned}\Re(p_{i,n}(t)) &= \frac{P_{\max}\overline{p}_{i,2n-1}(t)}{\sqrt{\psi_i(t)}},\\ \Im(p_{i,n}(t)) &= \frac{P_{\max}\overline{p}_{i,2n}(t)}{\sqrt{\psi_i(t)}},\end{aligned} \tag{13}$$

where $\psi_i(t) = ||\overline{\mathbf{p}}_i(t)||$. In addition, by applying the softmax function to the element of $\mathbf{c}(t)$, we define $\overline{\mathbf{c}}(t)$ as the normalized vector of $\mathbf{c}(t)$, where the $\overline{C_k}(t) \in \overline{\mathbf{c}}(t)$ is calculated as

$$\overline{C_k}(t) = \frac{e^{C_k(t)}}{\sum_{i=1}^{K} e^{C_i(t)}}. \tag{14}$$

Furthermore, we define $\overline{\phi_0}, \overline{\theta_0}, \overline{v_0} \in [0,1]$ as normalized variables of ϕ_0, θ_0, and v_0, respectively, to eliminate the effect of diversity of the variables. Thus, we have

$$\begin{aligned}\phi_0(t) &= \overline{\phi_0}(t)2\pi,\\ \theta_0(t) &= \overline{\theta_0}(t)\pi,\\ v_0(t) &= \overline{v_0}(t)v_{0,\max}.\end{aligned} \tag{15}$$

As a result, all the action variables are normalized in the range of $[0,1]$. The action $a(t)$ can be rewritten by

$$a(t) = \{\overline{\mathbf{P}}(t), \overline{\mathbf{c}}(t), \overline{\phi_0}(t), \overline{\theta_0}(t), \overline{v_0}(t)\}. \tag{16}$$

- Reward: The agent is given an immediate reward $r(t)$ upon performing action $a(t)$. This study aims to maximize the system sum-rate. Therefore, the reward is determined by the combined achievable rate of all GUs, which is represented as

$$r(t) = \sum_{k\in\mathcal{K}} R_k(t). \tag{17}$$

The agent aims to maximize the discounted cumulative reward, expressed as

$$R = \max_{a(t)} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r(t)\right]. \tag{18}$$

4.2. TD3-RFBS Optimization Framework

Since the action space is continuous and high-dimensional, the existing works use the DDPG algorithm as a DRL method for decision making [19,20,26,34,35]. However, DDPG is sensitive to the hyperparameters and DNN size and it often overestimates the Q-value, leading to suboptimal or unstable policies. Recently, the TD3 algorithm was proposed to tackle the above issues with three key techniques [36]:

- Clipped double Q-learning: TD3 uses two critic networks instead of one, as in DDPG. By using the minimum Q-value from the target networks, TD3 improves the accuracy of value estimates and reduces the overestimation bias.
- Target policy smoothing: TD3 adds noise to the target action when updating the policy. This makes the policy more robust to Q-value estimation errors.
- Delayed policy updates: TD3 updates the policy and target networks less frequently than the critic networks. This prevents the policy from exploiting the overestimated Q-values.

TD3 is a more stable and robust algorithm with these three improvements than DDPG. To address the defined MDP model, we propose the TD3-RFBS optimization framework, which leverages the advantages of TD3 over DDPG. TD3 contains three main DNNs: one actor network and two critic networks. The actor network selects actions based on the existing state, whereas the critic networks assess the actions produced by the actor network. Using two critic networks is intended to address the issue of overestimating the Q-values. Each main network is accompanied by a target network for stabilizing the training process. Accordingly, TD3 consists of six DNNs: an actor network $\mu(s|\theta^{\mu})$ with parameter θ^{μ}, two critic networks $Q_1(s,a|\theta^{Q_1})$ and $Q_2(s,a|\theta^{Q_2})$ with parameter θ^{Q_1} and θ^{Q_2}, respectively, a target actor network $\mu'(s|\theta^{\mu'})$ with parameter $\theta^{\mu'}$, and two target critic networks $Q_1'(s,a|\theta^{Q_1'})$ and $Q_2'(s,a|\theta^{Q_2'})$ with parameter $\theta^{Q_1'}$ and $\theta^{Q_2'}$, respectively. During the training procedure, the actor network $\mu(s|\theta^{\mu})$ is used to generate action $a(t)$ as

$$a(t) = \mu(s(t)|\theta^{\mu}) + \mathcal{N}(0,\sigma), \tag{19}$$

where $\mathcal{N}$ represents a noise following the Gaussian process. When the agent takes action $a(t)$, the environment changes from state $s(t)$ to state $s(t+1)$. The agent is given a reward $r(t)$ corresponding to the state–action pair $(s(t), a(t))$. This sample $(s(t), a(t), r(t), s(t+1))$ is saved in the experience replay buffer $\mathcal{R}$, which is utilized for updating the network parameters. A randomly sampled mini-batch of S transition tuples is extracted from $\mathcal{R}$ to update the actor and critic networks. The critic networks are updated by minimizing the loss function $L(\theta^{Q_j})_{j=1,2}$, given by

$$L(\theta^{Q_j}) = S^{-1}\sum_{i}^{S}(Y_i - Q_j(s_i, a_i|\theta^{Q_j}))^2, j = 1, 2, \tag{20}$$

$$Y_i = r_i + \gamma \min_{j=1,2} Q_j'\Big(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}) + \epsilon|\theta^{Q_j'}\Big), \epsilon \sim \text{clip}(\mathcal{N}(0,\sigma), -c, c), \tag{21}$$

where ϵ represents a smoothing noise and c is the binding of noise. TD3 employs delayed policy updates and utilizes the deterministic policy gradient to update the actor network parameters every f iterations, as

$$\nabla_{\theta^{\mu}} J(\theta^{\mu}) = S^{-1}\sum_{i}^{S} \nabla_a Q_1(s,a|\theta^{Q_1})|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}. \tag{22}$$

Lastly, TD3 employs a soft update approach to update the target network parameters. This is performed at a rate of $\rho \ll 1$ every f iterations and can be expressed as

$$\begin{aligned} \theta^{\mu'} &\leftarrow \rho\theta^{\mu} + (1-\rho)\theta^{\mu'}, \\ \theta^{Q_j'} &\leftarrow \rho\theta^{Q_j} + (1-\rho)\theta^{Q_j'}, j = 1, 2. \end{aligned} \tag{23}$$

Algorithm 1 describes the training process for the TD3-RFBS optimization framework. An actor network and two critic networks are initialized with random parameters, and corresponding target networks are created by copying the parameters from the original networks (lines 1–2). An experience replay buffer $\mathcal{R}$ is established to store the experience samples (line 3), with a predetermined capability that replaces the earliest sample with a new one upon reaching its limit. During each episode, an action $a(t)$ is generated from the current policy and noise given the current state $s(t)$, resulting in an immediate reward $r(t)$ and a next state $s(t+1)$ (lines 6–8). The sample $(s(t), a(t), r(t), s(t+1))$ is then stored in the buffer $\mathcal{R}$ (line 9). To train the networks, a mini-batch of S transitions is randomly sampled from the buffer (line 10). The parameters of the critic networks are updated using the loss function (lines 11–12). The actor network is updated using a delayed update strategy with deterministic gradient descent, and the target networks are updated using a soft update constant (lines 14–15). After the specified number of episodes, the training phase terminates, yielding a proficiently trained actor network (line 19). The trained actor network can then generate actions in real-time execution.

Algorithm 1 Training process for the TD3-RFBS optimization framework.

```
1:  Set up actor network $\mu(s|\theta^{\mu})$ and two critic networks $Q_j(s,a|\theta^{Q_j}), j = 1,2$ with parameters $\theta^{\mu}$
    and $\theta^{Q_j}, j = 1,2$
2:  Set up target networks $\mu'$ and $Q'_j, j = 1,2$ with parameters $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'_j} \leftarrow \theta^{Q_j}, j = 1,2$
3:  Establish experience replay buffer $\mathcal{R}$
4:  for each episode do
5:    for $t \in \{1, \ldots, T\}$ do
6:      Observe state $s(t)$
7:      Select action with exploration noise $a(t) = \mu(s(t)|\theta^{\mu}) + \mathcal{N}(0,\sigma)$
8:      Perform action $a(t)$, observe reward $r(t)$ and new state $s(t+1)$
9:      Save transition $(s(t), a(t), r(t), s(t+1))$ in $\mathcal{R}$
10:     Arbitrarily sample a mini-batch of $S$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $\mathcal{R}$
11:     $Y_i = r_i + \gamma Q'_j(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}) + \epsilon|\theta^{Q'_j}), \epsilon \sim \text{clip}(\mathcal{N}(0,\sigma), -c, c)$
12:     Update the critic networks: $L(\theta^{Q_j}) = S^{-1}\sum_i^S (Y_i - Q_j(s_i, a_i|\theta^{Q_j}))^2, j = 1,2$
13:     if $t$ mod $f$ then
14:       Update the actor network: $\nabla_{\theta^{\mu}} J = S^{-1}\sum_i^S \nabla_a Q_1(s,a|\theta^{Q_1})|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}$
15:       Update the target networks: $\theta^{\mu'} \leftarrow \rho\theta^{\mu} + (1-\rho)\theta^{\mu'}, \theta^{Q'_j} \leftarrow \rho\theta^{Q_j} + (1-\rho)\theta^{Q'_j}, j = 1,2$
16:     end if
17:   end for
18: end for
19: return well-trained actor network $\mu(s|\theta^{\mu})$.
```

4.3. Computational Complexity

In the following, the computational complexity of the TD3-RFBS framework is analyzed. According to [20,39], the computational complexity for a DNN is based on the number of multiplications as $\mathcal{O}\left(\sum_{l=0}^{L-1} n_l n_{l+1}\right)$, where L is the number of layers and n_l is the number of neurons of the l-th layer. In training mode, the TD3-RFBS algorithm uses a finite number of DNNs and requires $S \times M \times T$ iterations to complete the training phase, with S being the mini-batch size, M being the total count of episodes, and T being the number of steps per episode. As a result, the overall computational complexity can be estimated as $\mathcal{O}\left(SMT\sum_{l=0}^{L-1} n_l n_{l+1}\right)$.

In execution mode, the critic networks do not contribute to decision making. Instead, the FlyBS algorithm solely relies on the trained actor network to perform real-time decisions in a dynamic environment. The computational complexity associated with this process is approximately $\mathcal{O}\left(\sum_{l=0}^{L-1} n_l n_{l+1}\right)$.

5. Performance Evaluation and Discussion

In this section, we present comprehensive simulation results to demonstrate the effectiveness of our optimization framework. First, we describe the simulation settings. Then, we conduct a convergence analysis and compare the performance of our framework against several baseline methods.

5.1. Simulation Setup

We developed the RSMA-enhanced FlyBS communication system using Python (Version 3.10) and trained it using PyTorch (Version 1.12.1). We consider a FlyBS equipped with URA of N antennas. The FlyBS serves K single-antenna GUs randomly distributed in a 500 m × 500 m flat area. The initial FlyBS coordinate is set to $(500, 500, 100)$ m and the height range is limited to $[50, 200]$ m. The maximum speed of FlyBS and GUs are set to 20 and 5 m/s, respectively. The current simulation setup is limited in capturing the complexities of real-world environments, such as mountainous terrains. As an alternative, the Rician factor value is employed to control the communication channel quality, enabling a network performance assessment under varying channel conditions [26,34,35]. Unless otherwise specified, other parameters are set to their default values as shown in Table 2.

Table 2. Parameter setup.

Parameter	Value
System	
Number of GUs, K	8
Channel bandwidth, B	1 MHz
Noise power, σ^2	-174 dBm/Hz
GUs' maximum velocity	5 m/s
FlyBS's maximum velocity	20 m/s
FlyBS's maximum transmit power, $P_{\max}$	10 dBm
Number of antennas, N	16
Rician factor, κ	10
Large-scale path loss, L_k	$30 + 22\log(d_k)$ dB
Time slot duration, τ	0.1 s
Algorithm	
Optimizer	Adam
Discount factor, γ	0.95
Size of replay buffer	1×10^6
Size of mini-batch, S	64
Actor learning rate, lr_μ	1×10^{-3}
Critic learning rate, lr_Q	3×10^{-3}
Frequency of policy updates, f	2
Policy noise variance, σ	0.2
Noise clip, c	0.2
Soft update rate, ρ	1×10^{-3}
Number of training episodes	2000
Number of testing episodes	100
Number of time slots in each episode	300

To demonstrate the efficacy of the proposed framework, we compare it to several baseline algorithms, which are defined as follows.

- TD3-RFBS: Our proposed optimization framework is based on the TD3 algorithm with a normalized action space. It optimizes the precoding matrix, common rate vector, and 3D trajectory for the RSMA-enabled FlyBS to maximize the system sum-rate.
- TD3 algorithm for NOMA-based FlyBS (TD3-NFBS): TD3-NFBS uses the NOMA scheme for the communication channel between the FlyBS and GUs, as opposed to the RSMA scheme in TD3-RFBS. In NOMA, the user with the stronger signal must decode all other users' messages before accessing its own [7]. The TD3 algorithm optimizes the precoding matrix and 3D trajectory to maximize the system sum-rate.
- DDPG-based approach (DDPG) [20,26,34]: The formulated MDP model is solved using the well-known DDPG algorithm [33]. Action normalization is also applied to ensure fair comparisons.
- Local search-based approach (Local Search) [20,34]: An action is randomly generated within the feasible policy space. A local search algorithm then improves the action with a favorable reward at each step.

To assess the effectiveness of the methods, we use the sum-rate metric, which represents the total achievable rate of all users. A higher sum-rate indicates better performance. We run ten simulations for each method and compute the average values to ensure reliability.

5.2. Convergence Analysis

To assess the convergence of TD3-RFBS, we compare its convergence patterns of reward values using different learning rates. The learning rate is a critical parameter that significantly impacts the learning performance. We consider three sets of learning rates: $(lr_\mu, lr_Q) = \{(1 \times 10^{-2}, 3 \times 10^{-2}), (1 \times 10^{-3}, 3 \times 10^{-3}), (1 \times 10^{-4}, 3 \times 10^{-4})\}$, where lr_μ and lr_Q represent the learning rates of the actor and critic networks, respectively. All other hyperparameters are set to their default values. Figure 3 illustrates the convergence outcomes for the three sets of learning rates. Among them, the set $(lr_\mu, lr_Q) = (1 \times 10^{-3}, 3 \times 10^{-3})$ demonstrates the most favorable training performance in terms of higher rewards and stability. Consequently, this particular set of learning rates is chosen for the rest of the simulations.

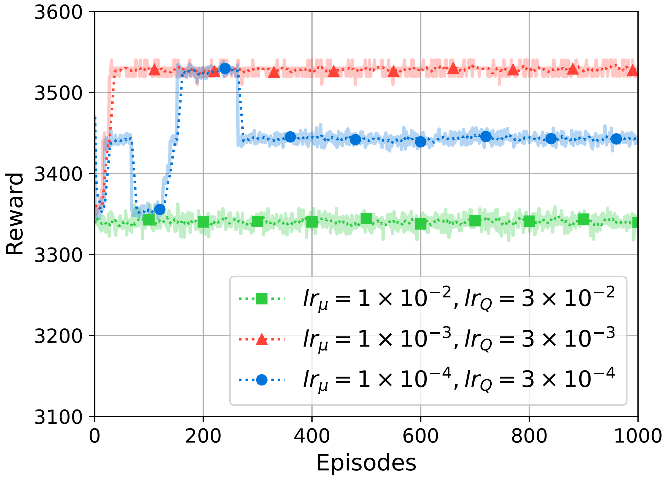

Figure 3. Convergence behavior of TD3-RFBS with three different learning rate configurations.

A comparative analysis is performed to evaluate the convergence of TD3-RFBS and DDPG, as illustrated in Figure 4. To ensure a fair comparison, the same hyperparameters are used for both methods, except those specific to TD3-RFBS. While TD3-RFBS and DDPG exhibit similar reward values at the initial stages, TD3-RFBS demonstrates superior convergence, achieving a higher and more stable reward after approximately 100 episodes. This is because DDPG is susceptible to overestimating value functions, which can lead to suboptimal policies. To mitigate this issue, TD3-RFBS employs two critic networks to generate two Q-value functions and uses the minimum Q-value during policy updates. It helps to reduce overestimation bias and improve the performance of TD3-RFBS.

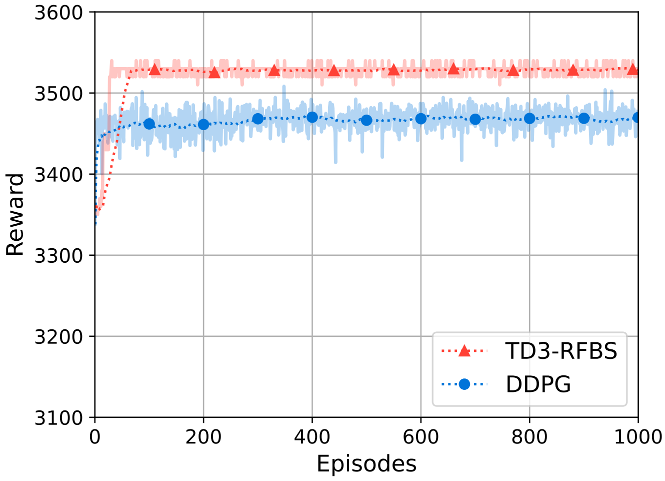

Figure 4. Convergence behavior of TD3-RFBS in comparison with DDPG.

5.3. Performance Analysis

In the following, to demonstrate the effectiveness of RSMA over NOMA, we assess the performance of TD3-RFBS and TD3-NFBS. We also investigate the impact of 3D and 2D FlyBS trajectories on performance. Finally, we compare the performance of TD3-RFBS, DDPG, and Local Search in the RSMA-enabled FlyBS system with a 3D trajectory by varying the transmission power and Rician factor.

5.3.1. Comparison of Multiple Access Schemes

Figure 5 compares the achievable sum-rate of the FlyBS system optimized by the TD3 algorithm using RSMA signaling (TD3-RFBS) and NOMA signaling (TD3-NFBS). As transmit power increases, both schemes exhibit an improved performance. However, RSMA performs better than NOMA due to rate-splitting and effective interference management. With NOMA, increased transmit power also increases interference, limiting the performance gains. Additionally, NOMA is complex due to the requirement for the user with the stronger signal to decode the messages of all other users, necessitating multi-layer SIC [10]. In contrast, RSMA achieves a superior performance to NOMA with a one-layer SIC, substantially reducing the complexity of the receiver. These results suggest that RSMA is a more effective signaling scheme for optimizing the system sum-rate in the FlyBS network.

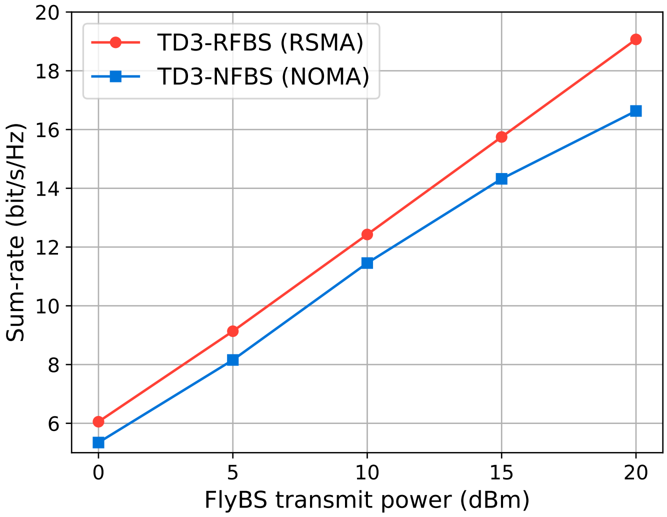

Figure 5. Sum-rate performance of the TD3-RFBS with its counterpart, TD3-NFBS, for different transmit power levels.

5.3.2. Comparison of 3D and 2D FlyBS Trajectories

In TD3-RFBS with 2D space (TD3-RFBS 2D), the FlyBS maintains a fixed altitude of 100 m, while in TD3-RFBS with 3D space (TD3-RFBS 3D), the FlyBS can fly at altitudes ranging from 50 to 200 m. The starting position of the FlyBS is set to $(500, 500, 100)$ m. Figure 6 illustrates an example of the FlyBS trajectories obtained by TD3-RFBS 3D and TD3-RFBS 2D. We can see that the FlyBS trajectory obtained by TD3-RFBS 3D is more complex than the FlyBS trajectory obtained by TD3-RFBS 2D. This is because TD3-RFBS 3D has the freedom to fly at different altitudes, which allows it to improve the channel quality. As a result, TD3-RFBS 3D is expected to outperform TD3-RFBS 2D regarding coverage, capacity, and reliability.

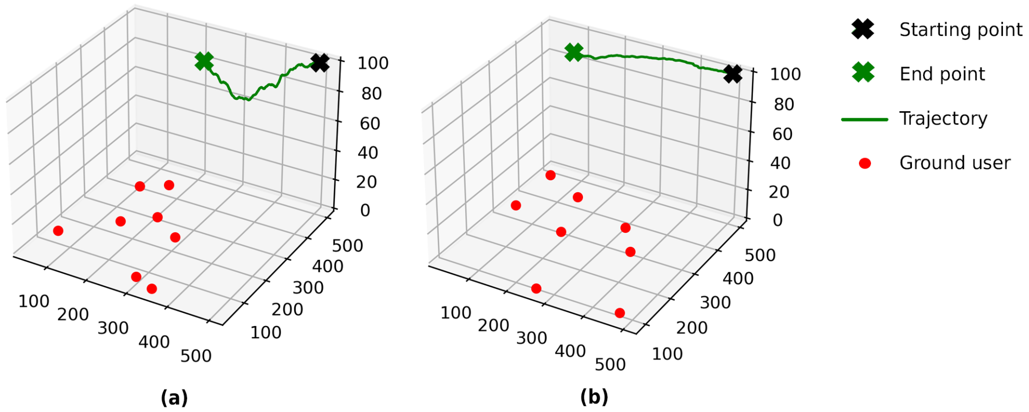

Figure 6. FlyBS trajectories obtained by TD3-RFBS. (**a**) 3D trajectory; (**b**) 2D trajectory.

Figure 7 shows the sum-rate achieved by TD3-RFBS 3D and TD3-RFBS 2D for different transmit power levels. As the transmission power increases, the system sum-rate increases proportionally. Notably, controlling the FlyBS altitude improves the system sum-rate. For instance, at a transmit power of 20 dBm, the sum-rate for TD3-RFBS 3D is 19.07 bit/s/Hz, which is a 15.09% improvement over the sum-rate of 16.57 bit/s/Hz for TD3-RFBS 2D. The key to TD3-RFBS 3D's superior performance is its ability to control the FlyBS altitude. Positioning the FlyBS at a suitable altitude can reduce interference and improve the LoS links to the GUs. In contrast, with a 2D trajectory as in TD3-RFBS 2D, the FlyBS cannot change its altitude in any channel conditions, which limits its performance.

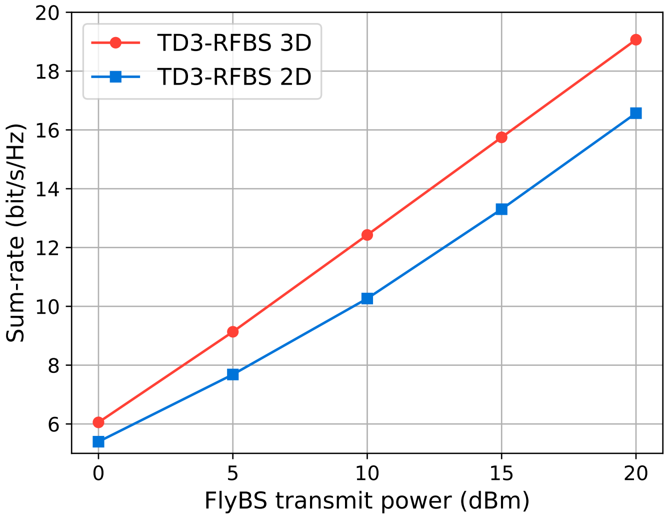

Figure 7. Sum-rate versus transmit power for 3D and 2D FlyBS trajectories using TD3-RFBS.

5.3.3. Comparison of Algorithms

We assess the effectiveness of TD3-RFBS against DDPG and Local Search algorithms across varying parameters in the network environment with the RSMA-enabled communications and 3D trajectory of the FlyBS. Figure 8 plots the downlink sum-rate achieved by TD3-RFBS, DDPG, and Local Search versus the transmit power of the FlyBS. As the FlyBS's transmission power escalates, the sum-rate of all methods also increases linearly. Compared to Local Search, DRL-based approaches consistently achieve a higher sum-rate. Local Search starts with an initial solution and iteratively improves it by making small changes; however, it can become stuck in the local optima and may not find the global optimum. One advantage of DRL approaches, TD3-RFBS, and DDPG over Local Search is their ability to learn from experience and improve their performance over time. Of the two compared DRL approaches, TD3-RFBS achieves better results than DDPG since TD3-RFBS overcomes the overestimation issue of DDPG in policy learning. At a transmit power of 20 dBm, TD3-RFBS obtains a sum-rate of 19.07 bit/s/Hz, which is 8.66% and 17.57% higher than the sum-rate values achieved by DDPG and Local Search, which are 17.55 and 16.22 bit/s/Hz, respectively.

Finally, the proposed framework and baseline schemes are evaluated in fading channels with various Rician factor values. Fading channels are suitable models for emulating realistic channel conditions in wireless communications, such as multipath scattering, temporal dispersion, and Doppler shifts that arise from the relative movement between the transmitter and receiver [26,34,35]. Figure 9 shows the achievable sum-rate results for five values of the Rician factor, $\kappa = \{10^{-1}, 10^{0}, 10^{1}, 10^{2}, \infty\}$. As the Rician factor diminishes in value, the NLoS element within the communication channel experiences an increase, rising uncertainty in the channel condition. In particular, when the Rician factor is ∞, it represents an ideal scenario without any chaotic signal, resulting in an optimal performance for all methods. We can observe that TD3-RFBS performs well in an environment with a Rician factor of 10^1, where the obtained sum-rate value is close to that of the ideal scenario. In contrast, DDPG and Local Search can only achieve results near the ideal situation when the Rician factor is up to 10^2. Overall, the sum-rate increases with the Rician factor value and our proposed framework, TD3-RFBS, outperforms the comparison schemes.

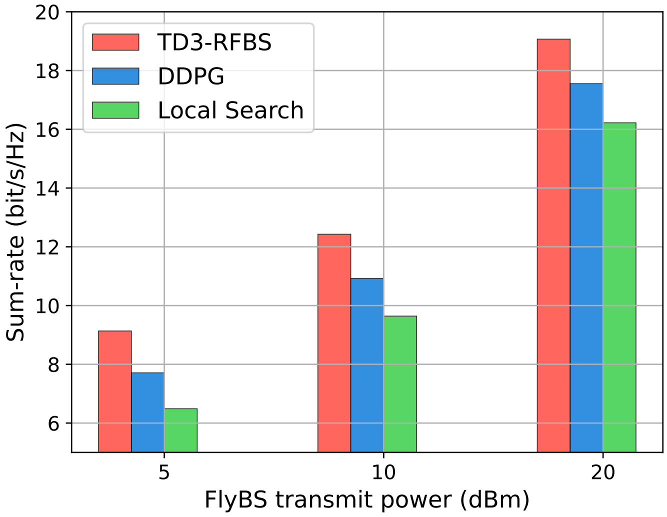

Figure 8. Sum-rate achieved by three compared methods for various transmit power levels of FlyBS.

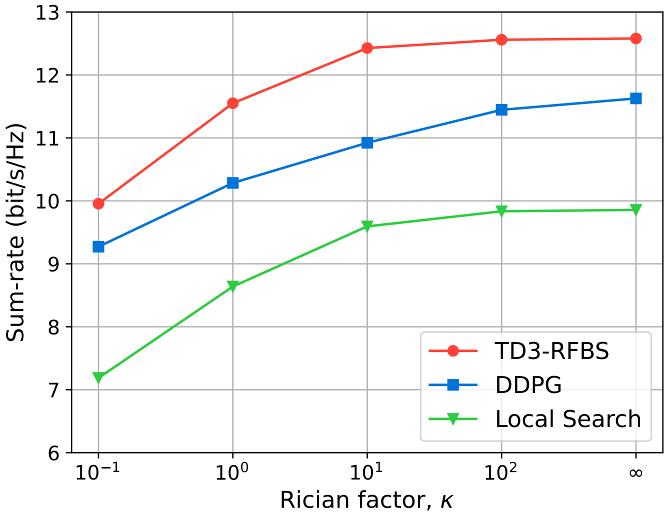

Figure 9. Sum-rate performance of three methods with varying Rician factor values.

5.4. Practical System Implication

Throughout the evaluations, TD3-RFBS efficiently optimizes the FlyBS trajectory and RSMA parameters for various configurations in the RSMA-enabled FlyBS system. This work has significant practical implications for future wireless networks, as it can improve the performance and reliability of aerial communication systems. This could benefit diverse applications such as disaster management, remote sensing, search and rescue, smart agriculture, and intelligent transportation systems [1,5,19–21]. Examples are as follows.

- Disaster management: FlyBSs can provide communication coverage in areas where terrestrial BSs have been damaged or destroyed by natural disasters, such as hurricanes and earthquakes. The TD3-RFBS optimization framework can improve the performance of these systems, leading to more reliable and efficient communications in disaster areas, which can be critical for search and rescue efforts.
- Remote sensing: FlyBSs can collect data on remote areas, such as forests, mountains, and oceans, for various applications, including environmental monitoring and remote control of critical infrastructure. The TD3-RFBS optimization framework can increase the spectral efficiency of these systems, leading to more timely and accurate data collection.

While the practical implementation of the proposed framework may face challenges such as hardware and software requirements, regulatory issues, and cost considerations, we believe that these challenges can be overcome through further research and development.

6. Conclusions

This paper investigated a sum-rate maximization problem in an RSMA-enhanced FlyBS system. Since the problem is non-convex and the TD3 algorithm is superior to the DDPG algorithm, we reformulated the problem as an MDP. We developed an optimization framework that utilizes TD3, namely TD3-RFBS, to solve the MDP model. This framework jointly optimizes the 3D FlyBS trajectory, precoding matrix, and common rate allocation without requiring prior knowledge of CSI. Simulation results showed that the TD3-based algorithm outperformed the DDPG-based algorithm in terms of reward and convergence. Moreover, the proposed framework achieved significant rate improvements compared to other baseline solutions in various scenarios with time-varying channel conditions.

In future work, the system model could be extended to include a multi-FlyBS environment, paving the way for developing multi-agent DRL methods to tackle the optimization challenge. To enhance the realism of the simulation environment, future work could focus on incorporating detailed terrain models that capture the impact of geographical features on communication channels. Additionally, exploring new communication technologies, such as reconfigurable intelligent surfaces and terahertz communications, could further enhance the system sum-rate. Finally, further research and development in practical systems are necessary to fully realize the potential of the proposed optimization framework.

Author Contributions: Conceptualization, T.-H.N. and L.P.; methodology, T.-H.N., L.V.N., L.M.D. and V.T.H.; software, T.-H.N. and L.V.N.; validation, T.-H.N. and L.V.N.; formal analysis, T.-H.N., L.M.D. and V.T.H.; investigation, T.-H.N. and L.P.; resources, L.P.; data curation, T.-H.N. and V.T.H.; writing—original draft preparation, T.-H.N.; writing—review and editing, L.V.N., L.M.D., V.T.H. and L.P.; visualization, T.-H.N. and L.V.N.; supervision, L.P.; project administration, L.P.; funding acquisition, T.-H.N. and L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by Seoul National University of Science and Technology.

Data Availability Statement: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two-dimensional
3D	Three-dimensional
BS	Base station
CSI	Channel state information
DDPG	Deep deterministic policy gradient
DNN	Deep neural network
DRL	Deep reinforcement learning
FlyBS	Flying base station
GU	Ground user
IoT	Internet of Things
LoS	Line-of-sight
MDP	Markov decision process
NLoS	Non-line-of-sight
NOMA	Non-orthogonal multiple access
QoS	Quality of service
RSMA	Rate-splitting multiple access
SIC	Successive interference cancellation
TD3	Twin-delayed deep deterministic policy gradient
UAV	Unmanned aerial vehicle
URA	Uniform rectangular array

References

1. Dao, N.N.; Pham, Q.V.; Tu, N.H.; Thanh, T.T.; Bao, V.N.Q.; Lakew, D.S.; Cho, S. Survey on Aerial Radio Access Networks: Toward a Comprehensive 6G Access Infrastructure. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1193–1225. [CrossRef]
2. Zhao, M.; Chen, C.; Liu, L.; Lan, D.; Wan, S. Orbital collaborative learning in 6G space-air-ground integrated networks. *Neurocomputing* **2022**, *497*, 94–109. [CrossRef]
3. Dong, F.; Song, J.; Zhang, Y.; Wang, Y.; Huang, T. DRL-Based Load-Balancing Routing Scheme for 6G Space–Air–Ground Integrated Networks. *Remote Sens.* **2023**, *15*, 2801. [CrossRef]
4. Lakew, D.S.; Tran, A.T.; Dao, N.N.; Cho, S. Intelligent Offloading and Resource Allocation in Heterogeneous Aerial Access IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 5704–5718. [CrossRef]
5. Geraci, G.; Garcia-Rodriguez, A.; Azari, M.M.; Lozano, A.; Mezzavilla, M.; Chatzinotas, S.; Chen, Y.; Rangan, S.; Renzo, M.D. What Will the Future of UAV Cellular Communications Be? A Flight From 5G to 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1304–1335. [CrossRef]
6. Shi, J.; Cong, P.; Zhao, L.; Wang, X.; Wan, S.; Guizani, M. A Two-Stage Strategy for UAV-enabled Wireless Power Transfer in Unknown Environments. *IEEE Trans. Mob. Comput.* 2023, *in press*. [CrossRef]
7. Dai, L.; Wang, B.; Yuan, Y.; Han, S.; Chih-lin, I.; Wang, Z. Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends. *IEEE Commun. Mag.* **2015**, *53*, 74–81. [CrossRef]
8. Mao, Y.; Dizdar, O.; Clerckx, B.; Schober, R.; Popovski, P.; Poor, H.V. Rate-Splitting Multiple Access: Fundamentals, Survey, and Future Research Trends. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2073–2126. [CrossRef]
9. Clerckx, B.; Mao, Y.; Jorswieck, E.A.; Yuan, J.; Love, D.J.; Erkip, E.; Niyato, D. A Primer on Rate-Splitting Multiple Access: Tutorial, Myths, and Frequently Asked Questions. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1265–1308. [CrossRef]
10. Mao, Y.; Clerckx, B.; Li, V.O. Rate-splitting multiple access for downlink communication systems: Bridging, generalizing, and outperforming SDMA and NOMA. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 133. [CrossRef]
11. Mao, Y.; Clerckx, B.; Li, V.O.K. Rate-Splitting for Multi-Antenna Non-Orthogonal Unicast and Multicast Transmission: Spectral and Energy Efficiency Analysis. *IEEE Trans. Commun.* **2019**, *67*, 8754–8770. [CrossRef]
12. Sen, S.; Santhapuri, N.; Choudhury, R.R.; Nelakuditi, S. Successive Interference Cancellation: Carving Out MAC Layer Opportunities. *IEEE Trans. Mob. Comput.* **2013**, *12*, 346–357. [CrossRef]
13. Hieu, N.Q.; Hoang, D.T.; Niyato, D.; Kim, D.I. Optimal Power Allocation for Rate Splitting Communications with Deep Reinforcement Learning. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 2820–2823. [CrossRef]
14. Xu, Y.; Mao, Y.; Dizdar, O.; Clerckx, B. Rate-Splitting Multiple Access with Finite Blocklength for Short-Packet and Low-Latency Downlink Communications. *IEEE Trans. Veh. Technol.* **2022**, *71*, 12333–12337. [CrossRef]
15. Van, N.T.T.; Luong, N.C.; Feng, S.; Nguyen, V.D.; Kim, D.I. Evolutionary Games for Dynamic Network Resource Selection in RSMA-Enabled 6G Networks. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1320–1335. [CrossRef]
16. Yu, D.; Kim, J.; Park, S.H. An Efficient Rate-Splitting Multiple Access Scheme for the Downlink of C-RAN Systems. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1555–1558. [CrossRef]

17. Xu, Y.; Mao, Y.; Dizdar, O.; Clerckx, B. Max-Min Fairness of Rate-Splitting Multiple Access with Finite Blocklength Communications. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6816–6821. [CrossRef]
18. Zhou, G.; Mao, Y.; Clerckx, B. Rate-Splitting Multiple Access for Multi-Antenna Downlink Communication Systems: Spectral and Energy Efficiency Tradeoff. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 4816–4828. [CrossRef]
19. Truong, T.P.; Dao, N.N.; Cho, S. HAMEC-RSMA: Enhanced Aerial Computing Systems with Rate Splitting Multiple Access. *IEEE Access* **2022**, *10*, 52398–52409. [CrossRef]
20. Nguyen, T.H.; Park, L. HAP-Assisted RSMA-Enabled Vehicular Edge Computing: A DRL-Based Optimization Framework. *Mathematics* **2023**, *11*, 2376. [CrossRef]
21. Jaafar, W.; Naser, S.; Muhaidat, S.; Sofotasios, P.C.; Yanikomeroglu, H. On the Downlink Performance of RSMA-Based UAV Communications. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16258–16263. [CrossRef]
22. Singh, S.K.; Agrawal, K.; Singh, K.; Li, C.P. Ergodic Capacity and Placement Optimization for RSMA-Enabled UAV-Assisted Communication. *IEEE Syst. J.* **2023**, *17*, 2586–2589. [CrossRef]
23. Singh, S.K.; Agrawal, K.; Singh, K.; Chen, Y.M.; Li, C.P. Performance Analysis and Optimization of RSMA Enabled UAV-Aided IBL and FBL Communication with Imperfect SIC and CSI. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 3714–3732. [CrossRef]
24. Xiao, M.; Cui, H.; Huang, D.; Zhao, Z.; Cao, X.; Wu, D.O. Traffic-Aware Energy-Efficient Resource Allocation for RSMA Based UAV Communications. *IEEE Trans. Netw. Sci. Eng.* 2023, *in press*. [CrossRef]
25. Ji, J.; Cai, L.; Zhu, K.; Niyato, D. Decoupled Association with Rate Splitting Multiple Access in UAV-assisted Cellular Networks Using Multi-agent Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* 2023, *in press*. [CrossRef]
26. Hua, D.T.; Do, Q.T.; Dao, N.N.; Cho, S. On sum-rate maximization in downlink UAV-aided RSMA systems. *ICT Express* 2023, *in press*. [CrossRef]
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
28. Sánchez, J.A.H.; Casilimas, K.; Rendon, O.M.C. Deep Reinforcement Learning for Resource Management on Network Slicing: A Survey. *Sensors* **2022**, *22*, 3031. [CrossRef]
29. Wang, Z.; Pan, W.; Li, H.; Wang, X.; Zuo, Q. Review of Deep Reinforcement Learning Approaches for Conflict Resolution in Air Traffic Control. *Aerospace* **2022**, *9*, 294. [CrossRef]
30. Nguyen, T.H.; Park, L. A Survey on Deep Reinforcement Learning-driven Task Offloading in Aerial Access Networks. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 822–827. [CrossRef]
31. Nguyen, T.H.; Park, H.; Park, L. Recent Studies on Deep Reinforcement Learning in RIS-UAV Communication Networks. In Proceedings of the 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, 20–23 February 2023; pp. 378–381. [CrossRef]
32. Nguyen, T.H.; Park, H.; Seol, K.; So, S.; Park, L. Applications of Deep Learning and Deep Reinforcement Learning in 6G Networks. In Proceedings of the 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), Paris, France, 4–7 July 2023; pp. 427–432. [CrossRef]
33. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016.
34. Truong, T.P.; Tuong, V.D.; Dao, N.N.; Cho, S. FlyReflect: Joint Flying IRS Trajectory and Phase Shift Design Using Deep Reinforcement Learning. *IEEE Internet Things J.* **2023**, *10*, 4605–4620. [CrossRef]
35. Hua, D.T.; Do, Q.T.; Dao, N.N.; Nguyen, T.V.; Lakew, D.S.; Cho, S. Learning-based Reconfigurable Intelligent Surface-aided Rate-Splitting Multiple Access Networks. *IEEE Internet Things J.* **2023**, *10*, 17603–17619. [CrossRef]
36. Fujimoto, S.; 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. 1587–1596.
37. Nguyen, T.H.; Truong, T.P.; Dao, N.N.; Na, W.; Park, H.; Park, L. Deep Reinforcement Learning-based Partial Task Offloading in High Altitude Platform-aided Vehicular Networks. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 1341–1346. [CrossRef]
38. Yong, S.K.; Thompson, J. Three-dimensional spatial fading correlation models for compact MIMO receivers. *IEEE Trans. Wirel. Commun.* **2005**, *4*, 2856–2869. [CrossRef]
39. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 15 July 2023).

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