FlyReflect: Joint Flying IRS Trajectory and Phase Shift Design using Deep Reinforcement Learning

Thanh Phung Truong, Van Dat Tuong, Nhu-Ngoc Dao, and Sungrae Cho

***Abstract*—Aerial access infrastructures have been considered a compulsory component of the sixth-generation (6G) networks, where airborne vehicles play the role of mobile access points to service ground users from the sky. In this scenario, intelligent reflecting surface (IRS) is one of the promising technologies associated with airborne vehicles for coverage extensions and throughput improvements, a.k.a., flying IRS. This study considers a multi-user multiple-input single-output (MISO) flying IRS system, where the flying IRS reflects downlink signals from ground base stations to users located at underserved areas where direct communications are unavailable. To achieve the system sum-rate maximization, we proposed a deep reinforcement learning (DRL) algorithm named `FlyReflect` to jointly optimize the flying trajectory and IRS phase shift matrix. First, end-to-end communications from a base station to its ground users via the flying IRS are analyzed to identify environmental and operational factors that impact achievable system sum-rate. Subsequently, the system is transformed into a DRL model, which is resolvable by the deep deterministic policy gradient (DDPG) algorithm. To improve the action decision accuracy of the DDPG algorithm, we proposed a mapping function to guarantee that all constraints are satisfied regardless of noise additions in the exploration process. Simulation results showed that our proposed algorithm outperforms state-of-the-art algorithms in multiple scenarios.**

***Index Terms*—Flying reflection, unmanned aerial vehicle, intelligent reflecting surface, deep reinforcement learning**

I. Introduction

THE boom of the Internet of things (IoT) era leads to a significant number of challenges ahead [1] in terms of data transfer rate, low latency, cost efficiency, broadband and spectrum efficiency, concurrent device connectivity, and energy efficiency, which mobile networks must address well to ensure successful implementation. To this end, the disruptive technologies in the sixth-generation (6G) networks, such as the terahertz spectrum, cell-less and aerial architectures, and massive distributed intelligence, are considered key enablers to support a unified access framework that harmonizes air-ground networks, large intelligent surface, and edge artificial intelligence (AI) [2].

As a foundational technology for such future communication, intelligent reflecting surface (IRS) [3]–[6], a.k.a. reconfigurable intelligent surface (RIS) [7]–[9] or large intelligent surface [10]–[12], which is constituted by multiple passive scattering elements in a two-dimensional (2D) surface, provides advantages in throughput improvement. In IRS, the features of reflecting elements are adjusted by a central software-defined controller to yield efficient signal reflection. In particular, the phases of every element are shifted to optimal values to reflect incoming signals, creating a positive multi-path effect [13].

In the context of 6G aerial radio access networks, IRS may associate with airborne vehicles to realize an efficient strategy for serving heterogeneous users from the sky [14], This association forms a new variance of the IRS class, i.e., flying IRS (F-IRS). An F-IRS consists of an airborne vehicle acting as a transporter that carries the IRS, which acts as a reflector. In such a system model, optimizing the three-dimensional (3D) flying trajectory and phase shift matrix at the F-IRS to obtain maximum average system sum-rate retains open challenges toward the maturation of the technology. In particular, to the best of our knowledge, although F-IRS systems have been partially investigated in several recent studies [15]–[26], no existing work has analyzed F-IRS optimization in a dynamic environment with a consideration of user mobility.

To tackle the aforementioned problem, we considered a system model where an F-IRS assists multi-user multiple-input single-output (MISO) communication to reflect downlink signals from a multi-antenna base station (BS) to their single-antenna ground users (GUs) located in underserved areas, where direct links between them are not available. In particular, GUs are assumed to have free mobility within the serving areas. In a nutshell, major contributions in this study are summarized as follows:

- First, we proposed a novel multi-user MISO F-IRS system model that considers user mobility. Here, the F-IRS reflects downlink signals from a multi-antenna BS to multiple single-antenna GUs moving around in the coverage area.
- Second, we formulated end-to-end communication channels from the BS to the GUs via F-IRS to identify possible environmental and operational factors that impact the achievable system sum-rate. Specifically, we developed mathematical expressions to represent the relationship between the 3D flying trajectory and the phase shift matrix of the F-IRS and the system sum-rate. Finally, an optimization utility of the achievable system sum-rate maximization is derived along with comprehensive constraints.
- Third, the optimization problem was transformed into a DRL model, which is resolvable by the deep deterministic policy gradient (DDPG) algorithm. To mitigate the effects of noise additions in the exploration process in the DDPG algorithm, we proposed a mapping function to guarantee that all constraints affected by the noise addition are

T. P. Truong, V. D. Tuong, and S. Cho are with the School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea. (emails: tptruong@uclab.re.kr, vdtuong@uclab.re.kr, srcho@cau.ac.kr)

N.-N. Dao is with the Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea. (email: nndao@sejong.ac.kr)

Corresponding author: Sungrae Cho.

satisfied to improve the action decision accuracy. The whole solution is referred to as `FlyReflect`.
- Fourth, we conducted Monte-Carlo experiments to analyze the impacts of hyperparameters on the proposed DRL framework in multiple scenarios with various environment parameter adjustments. Numerical results demonstrate that `FlyReflect` outperforms state-of-the-art algorithms in terms of system sum-rate achievement as well as convergence time.

The rest of this study is organized as follows. In Section II, we review some related works. Subsequently, the multi-user MISO F-IRS system model is analyzed and the optimization problem is then formulated in Section III. Next, we present our transformation of the system adopting the DRL model in Section IV to tackle the optimization problem. Then, we describe the performance experiments on the proposed framework in Section V. In section VI, conclusion of the paper is presented.

NOTATION: In this paper, for any general matrix $\boldsymbol{A}$, $\boldsymbol{A}^{\mathrm{H}}$ denotes the conjugate transpose of $\boldsymbol{A}$. The symbol $\otimes$ denotes the Kronecker product operation. For vector $\boldsymbol{a}$, the Euclidean norm is denoted as $||\boldsymbol{a}||$. The notation $|a|$ denotes the absolute value of a complex number. The circularly symmetric complex Gaussian distribution is denoted by $\mathcal{CN}(\chi, \sigma_0^2)$ with mean χ and variance σ_0^2.

II. Related Works

Recently, research on IRS has attracted significant attention from the academia and industry, especially in the context of beyond 5G and 6G communication networks. Generally, existing studies can be divided into two categories: fixed IRS schemes [15]–[22] and mobile IRS schemes [23]–[26].

Fixed IRS schemes consider the assistance of the IRS in a model, where the IRS is installed on fixed objects to improve system performance. In particular, the authors in [15] investigated the IRS support in a downlink multi-user communication system. They designed an iterative algorithm to optimize RIS configuration and digital beamforming in the BS for a sum-rate maximization problem. The IRS assists in a non-orthogonal multiple access (NOMA) network is considered in paper [16]. The authors examined an IRS-empowered downlink system to minimize the BS transmit power; for this purpose, they proposed a novel algorithm, namely a difference-of-convex programming algorithm, to jointly design the IRS phase shift matrix and BS transmit beamforming vector. In [17], the downlink transmission in a MISO NOMA system was investigated with the aid of the IRS. In this work, the authors formulated a problem that jointly optimized the passive and active beamforming vectors for the sum-rate maximization. They decoupled the problem into two sub-problem, then a successive convex approximation technique was applied for solving them.

MISO communication was also considered in many IRS studies [18]–[20]. Specifically, the authors in [18] examined the support of the IRS in an MISO network with eavesdroppers and a friendly jammer. They aimed to maximize the energy efficiency by jointly optimizing the IRS phase shift matrix, transmit beamforming, and jamming vectors. Then, they designed a semidefinite programming relaxation-based algorithm and an alternate optimization algorithm to handle the optimization problem. In [19], the RIS-based downlink transmission was considered in a multi-user MISO system. In this work, two algorithms based on sequential fractional programming and gradient descent were designed to optimize the transmit power allocation and the IRS phase shifts for the energy efficiency maximization problem. The downlink MISO communication with the help of RIS in [20] was considered in an unmanned aerial vehicle (UAV)-enabled wireless network system. The authors jointly designed the dynamic decoding order, IRS phase shift design, power allocation, and UAV trajectory to minimize the energy consumption using a decay deep Q-network is designed to handle this work. The fixed IRS in a UAV network has also appeared in many works such as [21], [22]. The authors in [21] proposed a scheme in which an RIS built on the wall of a building assists the UAV relaying system to improve the coverage, capacity, and bit error rate (BER) of the systems. The assistance of the IRS in UAV-supported terahertz communications was investigated in [22]. In this, the authors proposed a successive convex approximation with the rate constraint penalty-based algorithm to jointly design the allocation of the terahertz sub-bands, IRS phase shift, power control, and UAV trajectory for a minimum average achievable rate maximization problem.

On the other hand, the mobile IRS schemes, which were considered in numerous studies [23]–[26], have flexibility in deployment. As in [23], an aerial IRS (AIRS) was developed to aid a terrestrial communication system in order to maximize the minimum signal-to-noise ratio at a location. A closed-form optimal solution is presented for the AIRS placement, phase shift matrix, and transmit beamforming vector optimization. A UAV-borne reflecting surface system was investigated to assist the communication from a source to a destination in [24]. The authors analyzed the system performance according to ergodic capacity, outage probability, and energy efficiency in three modes, i.e., integrated UAV-IRS mode, IRS-only mode, and UAV-only mode. The AIRS in [25] was developed in a cell-free massive multiple input multiple output (MIMO) system to maximize the achievable rate of a single user. In this work, the authors presented a fast optimal search algorithm to jointly optimize the precoding vector at the access points, the transmit power allocation vector, the AIRS placement, and the phase shift matrix. An ARIS-assisted IoT wireless network was studied in [26]. The authors presented a problem for the expected sum age-of-information minimization by optimizing the UAV altitude, RIS phase shift elements, and communication schedule; then, a DRL approach was used to tackle the problem.

Although the above-mentioned mobile IRS studies have achieved some certain achievements in pointing out the advantages of the mobile IRS over the fixed IRS and the performance improvement of the system with the assistance of the IRS, no research has considered the mobile IRS in a multi-user dynamic environment with user mobility. This is the motivation for us to conduct this study, which investigates the effects of the mobile IRS, i.e., F-IRS, on the dynamic

environments where the users move around in the coverage area.

III. Problem Statement

This section proposed the achievable system sum-rate optimization problem, where the F-IRS system scenario and communication channel model are thoroughly examined. Table I lists the notations utilized in this study.

TABLE I: List of notations

Notation	Description
M	Number of BS antennas
K	Number of ground users (GUs)
N	Number of IRS elements
$\boldsymbol{H_T}$	Channel gain matrix of B2F link
$\boldsymbol{h_{R,k}}$	Channel gain matrix of F2k link
$\boldsymbol{\Theta}$	Phase shift matrix
y_k	Received signal at GU k
$\boldsymbol{w_k}$	Transmit beamforming vector
$\boldsymbol{H_T^{LoS}}$ / $\boldsymbol{H_T^{NLoS}}$	LoS / NLoS components of $\boldsymbol{H_T}$
$\boldsymbol{H_{R,k}^{LoS}}$ / $\boldsymbol{H_{R,k}^{NLoS}}$	LoS / NLoS components of $\boldsymbol{H_{R,k}}$
L_1 / $L_{2,k}$	Path losses of the B2F / F2k links
ε_1/$\varepsilon_{2,k}$	Rician factors of the B2F / F2k links
p_I / p_{BS} / p_k	Coordinates of F-IRS / BS antennas / GU k
d_{B2F} / d_{F2k}	Distance of the B2F / F2k links
$\varphi_{AoD,T}$ / $\phi_{AoD,T}$	Azimuth / elevation angle of departures (AoDs) at BS antennas to F-IRS
$\varphi_{AoA,T}$ / $\phi_{AoA,T}$	Azimuth / elevation angle of arrivals (AoAs) at F-IRS from BS
$\varphi_{AoD,Rk}$ / $\phi_{AoD,Rk}$	Azimuth / elevation angle of departures (AoDs) at F-IRS to GU k
a_M	Received array response vector at F-IRS
a_N / $a_{R,k}$	Transmitted array response vectors at BS antennas / F-IRS
$a_{NP}, a_{NQ}, a_{MU}, a_{MV}, a_{Rk,P}, a_{Rk,Q}$	Temporary vectors for presentation
ρ_N, ζ_N, ρ_M, ζ_M, ρ_R, ζ_R	Temporary parameters for presentation
d_u / d_v	Distance between two array elements of BS antennas with respect to z-axis / y-axis
d_p / d_q	Distance between two array elements of IRS with respect to y-axis / x-axis
R_k	Achievable rate at GU k
$[z_{min}, z_{max}]$	Altitude range of F-IRS
$t_u(t)$, $\varphi_u(t)$, $\phi_u(t)$	Movement variables of F-IRS

A. System Model

Fig. 1 illustrates the multi-user MISO F-IRS system model in consideration, which consists of an M-antenna BS and K ($K \geq 2$) single-antenna GUs, where direct links between the BS and GUs are unavailable due to obstacle. The communications between the BS and the GUs are through an F-IRS, which has N reflecting elements. To facilitate forthcoming analyses, a relative coordinate system is assumed at the location of the BS.

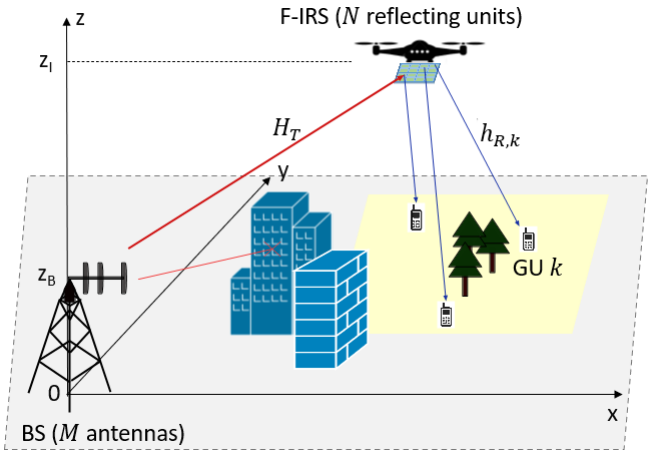


Fig. 1: Multi-user MISO F-IRS systems.

This study considers the downlink transmission, in which direct transmissions from the BS to GUs are blocked by obstructions; thus, there is no signal path of the direct channel. The communication channel from the BS to GU k is assisted by the F-IRS, which is constituted by three components: the transmission channel between the BS and the F-IRS (i.e., B2F) link, reflection at the F-IRS, and the F-IRS to the GU k reflection channel (i.e., F2k). We denote $\boldsymbol{H_T} \in \mathbb{C}^{N\times M}$ as the channel gain matrix of the B2F link, $\boldsymbol{h_{R,k}} \in \mathbb{C}^{1\times N}$ as the reflection channel gain vector of the F2k link. The phase shift matrix at the F-IRS is defined by a diagonal matrix $\boldsymbol{\Theta} = \text{diag}\left(\alpha_1 e^{j\theta_1}, \cdots, \alpha_n e^{j\theta_n}, \cdots, \alpha_N e^{j\theta_N}\right)$, where each reflecting element n has its reflection coefficient α_n in the range of $[0, 1]$ and the phase shift value $\theta_n \in [0, 2\pi]$, $\boldsymbol{\Theta} \in \mathbb{C}^{N\times N}$. Here, j denotes the imaginary unit. Hence, the received signals at GU k are given as

$$y_k = (\boldsymbol{h}_{R,k}\boldsymbol{\Theta}\boldsymbol{H}_T)\boldsymbol{x} + n_k, \tag{1}$$

where $n_k \sim \mathcal{CN}(0, \sigma_0^2)$ denotes the additive white Gaussian noise (AWGN) received at the GU k, and $\boldsymbol{x} \in \mathbb{C}^{M\times K}$ is the transmitted signal at the BS, which is defined as

$$\boldsymbol{x} = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{s}_k, \tag{2}$$

where $\boldsymbol{s}_k$ denotes the transmit data symbol to GU k and $\boldsymbol{w}_k \in \mathbb{C}^{M\times 1}$ is the transmit beamforming vector.

B. Channel Model

Channels $\boldsymbol{H_T}$ and $\boldsymbol{h_{R,k}}$ are assumed following a well-known Rician fading [27]–[29]. Accordingly, the channels are modeled as

$$\begin{aligned} \boldsymbol{H_T} &= L_1\left(\sqrt{\frac{\varepsilon_1}{\varepsilon_1+1}}\boldsymbol{H_T^{LoS}} + \sqrt{\frac{1}{\varepsilon_1+1}}\boldsymbol{H_T^{NLoS}}\right), \\ \boldsymbol{h_{R,k}} &= L_{2,k}\left(\sqrt{\frac{\varepsilon_{2k}}{\varepsilon_{2k}+1}}\boldsymbol{h_{R,k}^{LoS}} + \sqrt{\frac{1}{\varepsilon_{2k}+1}}\boldsymbol{h_{R,k}^{NLoS}}\right), \end{aligned} \tag{3}$$

where the Rician factors of the B2F and F2k links are ε_1 and ε_{2k}, respectively, L_1 and $L_{2,k}$ are the corresponding path losses, $\boldsymbol{H_T^{LoS}}$ and $\boldsymbol{h_{R,k}^{LoS}}$ are the respective light of sight (LoS)

components of the channels, and $\boldsymbol{H}_{\boldsymbol{T}}^{NLoS}$ and $\boldsymbol{h}_{\boldsymbol{R,k}}^{NLoS}$ are the respective non-light of sight (NLoS) components of the channels. In practice, the Rician fading channel is in the form of an LoS channel when the Rician factors equal ∞. In contrast, when the factors equal 0, the channel is in the form of an NLoS channel.

The NLoS components of the channels are given by $\mathcal{CN}(0,1)$. For the LoS components, we investigate two uniform rectangular arrays (URAs): (*i*) the antennas placed on the y-z plane at the BS, which has dimensions of U rows $\times$ V columns, resulting in $M = U \times V$; and (*ii*) the reflecting elements placed on the x-y plane of the F-IRS, which has dimensions of P rows $\times$ Q columns, resulting in $N = P \times Q$. In addition, we choose the bottom left element ($[u, v] = 0$) as the reference point of the BS's antennas and the bottom left element ($[p, q] = 0$) as the reference point of the F-IRS. The coordinates of the F-IRS and the antennas are denoted as $p_I = [x_I, y_I, z_I]$ and $p_{BS} = [0, 0, z_B]$, respectively, where z_B is the fixed altitude of the antenna at the BS. $[x_I, y_I, z_I]$ depends on the F-IRS's movement. We denote $p_k = [x_k, y_k, 0]$ as the coordinate of the GU k. Then, we denote d_{B2F} and d_{F2k} as the distances of the B2F and the F2k links, respectively, which are calculated as

$$\begin{aligned} d_{B2F} &= \sqrt{(x_I^2 + y_I^2) + (z_I - z_B)^2}, \\ d_{F2k} &= \sqrt{(x_I - x_k)^2 + (y_I - y_k)^2 + z_I^2}. \end{aligned} \tag{4}$$

In addition, to determine the propagation direction of the channel, we denote $(\varphi_{AoD,T}, \phi_{AoD,T})$ as the azimuth angle of departures (AoDs) and elevation AoDs at the BS antennas to the F-IRS, $(\varphi_{AoA,T}, \phi_{AoA,T})$ as the azimuth angle of arrivals (AoAs) and elevation AoAs at the F-IRS from the BS, and $(\varphi_{AoD,Rk}, \phi_{AoD,Rk})$ as the azimuth AoDs and elevation AoDs at the F-IRS to the GU k, respectively. Then, the LOS components are calculated as

$$\begin{aligned} \boldsymbol{H}_{\boldsymbol{T}}^{LoS} &= a_N^{\mathrm{H}}(\phi_{AoA,T}, \varphi_{AoA,T}) a_M(\phi_{AoD,T}, \varphi_{AoD,T}), \\ \boldsymbol{h}_{\boldsymbol{R,k}}^{LoS} &= a_{R,k}(\phi_{AoD,Rk}, \varphi_{AoD,Rk}), \end{aligned} \tag{5}$$

where $a_M \in \mathbb{C}^{1\times M}$ denotes the received array response vector of the F-IRS, and $a_{R,k} \in \mathbb{C}^{N\times 1}$ and $a_N \in \mathbb{C}^{1\times N}$ denote the transmitted array response vector of the F-IRS and the BS antennas, respectively. The channel propagation model is illustrated in Fig. 2, and the response arrays are calculated as [30], [31]

$$a_N(\phi_{AoA,T}, \varphi_{AoA,T}) = a_{NP}(\rho_N) \otimes a_{NQ}(\zeta_N), \tag{6}$$

where

$$\begin{aligned} a_{NP}(\rho_N) &= [1, e^{j\rho_N}, ..., e^{j(P-1)\rho_N}], \\ a_{NQ}(\zeta_N) &= [1, e^{j\zeta_N}, ..., e^{j(Q-1)\zeta_N}], \\ \rho_N &= \frac{2\pi}{\lambda} d_p \sin\varphi_{AoA,T} \sin\phi_{AoA,T} = -\frac{2\pi}{\lambda} d_p \frac{y_I}{d_{B2F}}, \\ \zeta_N &= \frac{2\pi}{\lambda} d_q \cos\varphi_{AoA,T} \sin\phi_{AoA,T} = -\frac{2\pi}{\lambda} d_q \frac{x_I}{d_{B2F}}. \end{aligned}$$

Similarly,

$$a_M(\phi_{AoD,T}, \varphi_{AoD,T}) = a_{MU}(\rho_M) \otimes a_{MV}(\zeta_M), \tag{7}$$

where

$$\begin{aligned} a_{MU}(\rho_M) &= [1, e^{j\rho_M}, ..., e^{j(U-1)\rho_M}], \\ a_{MV}(\zeta_M) &= [1, e^{j\zeta_M}, ..., e^{j(V-1)\zeta_M}], \\ \rho_M &= \frac{2\pi}{\lambda} d_u \cos\phi_{AoD,T} = \frac{2\pi}{\lambda} d_u \frac{z_I - z_B}{d_{B2F}}, \\ \zeta_M &= \frac{2\pi}{\lambda} d_v \sin\varphi_{AoD,T} \sin\phi_{AoD,T} = \frac{2\pi}{\lambda} d_v \frac{y_I}{d_{B2F}}, \end{aligned}$$

and

$$a_{Rk}(\phi_{AoD,Rk}, \varphi_{AoD,Rk}) = a_{Rk,P}(\rho_R) \otimes a_{Rk,Q}(\zeta_R), \tag{8}$$

where

$$\begin{aligned} a_{Rk,P}(\rho_R) &= [1, e^{j\rho_R}, ..., e^{j(P-1)\rho_R}], \\ a_{Rk,Q}(\zeta_R) &= [1, e^{j\zeta_R}, ..., e^{j(Q-1)\zeta_R}], \\ \rho_R &= \frac{2\pi}{\lambda} d_p \sin\varphi_{AoD,Rk} \sin\phi_{AoD,Rk} = \frac{2\pi}{\lambda} d_p \frac{y_k - y_I}{d_{F2k}}, \\ \zeta_R &= \frac{2\pi}{\lambda} d_q \cos\varphi_{AoD,Rk} \sin\phi_{AoD,Rk} = \frac{2\pi}{\lambda} d_q \frac{x_k - x_I}{d_{F2k}}. \end{aligned}$$

Here, λ is the wavelength, d_u and d_v are the distances between the two array elements of the BS's antenna with respect to the z-axis and y-axis, respectively, and d_p and d_q are the distances between the two array elements of the IRS with respect to the y-axis and x-axis, respectively, a_{NP}, a_{NQ}, a_{MU}, a_{MV}, $a_{Rk,P}$, $a_{Rk,Q}$ and ρ_N, ζ_N, ρ_M, ζ_M, ρ_R, ζ_R are the temporary vectors and parameters for presentation. As a result, the communication channel from BS to GU k varies according to the coordinates of the F-IRS and the GU k, and the phase shift matrix. Therefore, the received signals at GU k are represented with the variables as

$$y_k(p_I, \boldsymbol{\Theta}, p_k) = \boldsymbol{h}_{R,k}(p_I, p_k) \boldsymbol{\Theta} \boldsymbol{H}_T(p_I) \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{s}_k + n_k. \tag{9}$$

Then, the achievable signal-to-interference-plus-noise ratio (SINR) at the GU k is calculated as

$$SINR_k = \frac{|\boldsymbol{h}_{R,k}(p_I, p_k) \boldsymbol{\Theta} \boldsymbol{H}_T(p_I) \boldsymbol{w}_k|^2}{\sum_{i\neq k}^{K} |\boldsymbol{h}_{R,k}(p_I, p_k) \boldsymbol{\Theta} \boldsymbol{H}_T(p_I) \boldsymbol{w}_i|^2 + \sigma_k^2}. \tag{10}$$

Accordingly, the corresponding achievable rate at GU k is calculated as

$$R_k = \Omega_k log_2(1 + SINR_k), \tag{11}$$

where Ω_k is the corresponding usage bandwidth of GU k.

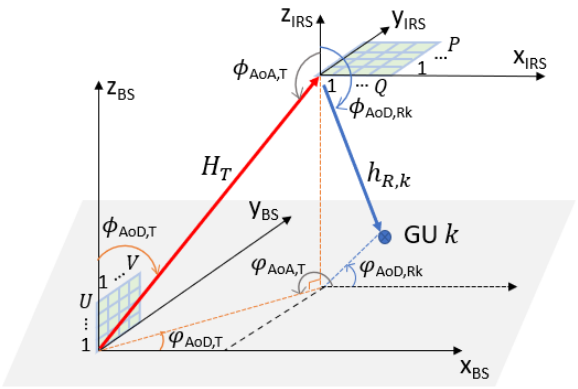


Fig. 2: Channel propagation model.

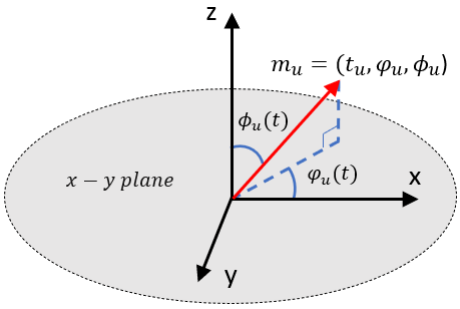


Fig. 3: F-IRS moving design.

C. Problem Formulation

Achievable system sum-rate (ASSR) maximization problem is formulated as follows. Because the positions of GUs are assumed stochastic, the phase shift matrix and the flying trajectory of the F-IRS are considered for the optimization. The F-IRS is assumed to move with a constant velocity v_c or hover in the air with a range of altitude at $[z_{min}, z_{max}]$, the positions of the F-IRS at time slot t and $(t+1)$ are $p_I(t)$ and $p_I(t+1)$, respectively. At each time slot period Δ_t, it can move with the maximum distance as $D_{\max} \triangleq v_c \Delta_t$. The IRS phase shift elements are designed for maximizing the reflection signal in practice; therefore, we set $\alpha_n = 1, \forall n$ [32]. Then, the ASSR optimization problem in this paper is formulated with the constraints as

$$\begin{aligned}
(\mathrm{P1}): \quad & \max_{p_I(t+1), \boldsymbol{\Theta}} \quad \sum_{k=1}^{K} R_k, && \text{(12a)}\\
\text{s.t.} \quad & 0 \le \theta_n \le 2\pi, \quad \forall n, && \text{(12b)}\\
& z_{min} \le z_I(t+1) \le z_{max}, \quad \forall t, && \text{(12c)}\\
& ||p_I(t+1) - p_I(t)||^2 \le D_{\max}^2, \quad \forall t, && \text{(12d)}
\end{aligned}$$

where constraint (12b) is the phase shift value requirement, constraint (12c) indicates the limited range of the F-IRS's height, and constraint (12d) ensures the maximum movement distance in a time slot of the F-IRS.

As shown in Fig. 3, we map the 3D movement of the F-IRS $m_u(t)$ into a vector with three variables as $(t_u(t), \varphi_u(t), \phi_u(t))$, where $t_u(t)$ is the time that the F-IRS moving in the time slot t, $\varphi_u(t)$ and $\phi_u(t)$ are the azimuth and elevation angles that represent the movement direction of the F-IRS at time slot t, respectively, with $0 \le t_u(t) \le \Delta_t$, $0 \le \varphi_u(t) \le 2\pi$, and $0 \le \phi_u(t) \le \pi$. After each time slot, the position of the F-IRS will be changed based on the movement, and the moving distance is calculated as

$$\begin{aligned}
x_{move}(t) &= v_c t_u(t) \sin(\phi_u(t)) \cos(\varphi_u(t)),\\
y_{move}(t) &= v_c t_u(t) \sin(\phi_u(t)) \sin(\varphi_u(t)),\\
z_{move}(t) &= v_c t_u(t) \cos(\phi_u(t)).
\end{aligned} \quad \text{(13)}$$

Then, the coordinate entries of $p_I(t+1)$ are calculated as

$$\begin{aligned}
x_I(t+1) &= x_I(t) + x_{move}(t),\\
y_I(t+1) &= y_I(t) + y_{move}(t),\\
z_I(t+1) &= z_I(t) + z_{move}(t),
\end{aligned} \quad \text{(14)}$$

and the constraints (12d) and (12c) is expressed as

$$0 \le t_u(t) \le \Delta_t, \ \forall t, \quad \text{(15)}$$

$$\frac{z_{min} - z_I(t)}{v_c} \le t_u(t) \cos(\phi_u(t)) \le \frac{z_{max} - z_I(t)}{v_c}, \ \forall t. \quad \text{(16)}$$

Accordingly, the problem (P1) is formulated as

$$\begin{aligned}
(\mathrm{P2}): \quad & \max_{\substack{t_u(t), \varphi_u(t), \phi_u(t),\\ \theta_n(t), n \in N}} \quad \sum_{k=1}^{K} R_k(t), && \text{(17a)}\\
\text{s.t.} \quad & 0 \le \theta_n(t) \le 2\pi, \quad \forall n, && \text{(17b)}\\
& 0 \le \varphi_u(t) \le 2\pi, \quad 0 \le \phi_u(t) \le \pi, \quad \forall t, && \text{(17c)}\\
& (15), (16) && \text{(17d)}
\end{aligned}$$

D. Markov Decision Process Transformation

It is observed that (P2) is a non-convex optimization problem, where finding an optimization solution using traditional optimization approaches is difficult and complicated. In particular, existing traditional algorithms that solved IRS phase shift optimization problems demand a high level of computer complexity [33]. Therefore, constructing an approximation method to discover the sub-optimal solution can be considered a viable alternative. Fortunately, reinforcement learning has recently emerged as an effective option adopting this strategy [34]–[38]. Inspired by this observation, we transform (P2) into a Markov decision process (MDP)-based problem and apply a reinforcement learning algorithm to resolve it.

The MDP is a stochastic control process that models decision-making by learning from interactions in order to achieve the desired outcome. The interactions are conducted by an agent with the role of the decision-maker and an environment the agent interacts [39]. The MDP is represented by a tuple of 4 elements including state, action, distribution probability, and reward. At each time slot, the state $s(t)$ contains the current position of the F-IRS and all the GUs, which is given as

$$s(t) = \{p_I(t), p_1(t), p_2(t), \ldots, p_K(t)\}. \quad \text{(18)}$$

For each state $s(t)$, the agent decides a feasible action $a(t)$ that contains optimization variables, which is given as

$$a(t) = \{t_u(t), \phi_u(t), \varphi_u(t), \theta_1(t), \theta_2(t), \ldots, \theta_N(t)\}. \quad \text{(19)}$$

The decided actions are obtained from policy $\boldsymbol{\pi}(t) : s(t) \rightarrow a(t)$, which is trained by reinforcement learning algorithms. By executing the action, the state $s(t)$ transfers to the next state $s(t+1)$ following the probability $P(s(t+1)|s(t), a(t))$. Then, the immediate reward received from the environment is calculated as the ASSR at time slot t

$$r(t|s(t), a(t)) = \sum_{k=1}^{K} R_k(t). \quad \text{(20)}$$

Consequently, the MDP-based problem of the system is formulated as

$$(\text{P3}): \quad \max_{a(t)} \quad \sum_{t=0}^{T-1} \gamma^t r(t|s(t), a(t)), \tag{21a}$$

$$\text{s.t.} \quad (17b), (17c), (17d), \tag{21b}$$

where γ is the discount factor, T is the number of considered time slots. The problem (P3) aims to maximize the long-term return of the system by optimizing the MDP action $a(t)$ at each observation state $s(t)$ while satisfying environment constraints.

IV. FlyReflect Framework

Due to the limited computation and energy resources in the F-IRS, the learning process is implemented in the BS. Therefore, the agent in this study is the BS, and the environment is the whole system. After receiving the states of the environment, the BS decides the action according to policy and sends it to the F-IRS through a wireless control link, the F-IRS then acts to the environment according to the control signals from the BS, the environment later sends the reward back to the agent to update the policy. Owing to the dynamic environment, the state and action have continuous values, which becomes the challenge of this reinforcement learning model. To tackle this issue, we develop a continuous DRL framework based on an actor-critic algorithm, namely DDPG.

A. Deep Deterministic Policy Gradient Algorithm

At first, we briefly introduce the DDPG algorithm, which is a combination of using the deep Q network (DQN) structure and the policy gradient (PG) algorithm [40]. The DDPG includes an actor network $\mu(s|\theta^\mu)$ that maps a state s to an action a in each time slot according to the policy, and a critic network $Q(s, a|\theta^Q)$ that measures the performance of the chosen action, in which θ^μ and θ^Q are the weight parameters of the actor and critic networks, respectively.

The critic network parameter is updated by minimizing the critic loss function as

$$L(\theta^Q) = [(Q(s(t), a(t)|\theta^Q) - y(t))^2], \tag{22}$$

where $Q(s(t), a(t)|\theta^Q)$ is the value of the action $a(t)$ obtained at the state $s(t)$, and $y(t)$ is the update target value. The value of $y(t)$ is expressed by

$$y(t) = r(t) + \gamma Q(s(t+1), \mu(s(t+1))|\theta^Q), \tag{23}$$

where the discount factor $\gamma \in [0, 1]$.

The actor network parameter is updated based on the critic network, using the policy's updating gradient as

$$\nabla_{\theta^\mu} J = \nabla_{\theta^\mu} Q(s(t), \mu(s(t)|\theta^\mu)|\theta^Q). \tag{24}$$

To improve stability in learning, the DDPG algorithm uses two target networks, the target actor and critic networks, which are denoted as $\mu'(s|\theta^{\mu'})$ and $Q'(s, a|\theta^{Q'})$, respectively. Then, the target value (23) is represented as

$$y(t) = r(t) + \gamma Q'(s(t+1), \mu'(s(t+1)|\theta^{\mu'})|\theta^{Q'}). \tag{25}$$

The target networks are updated via "soft updates" with a small constant $\urcorner$ ($\urcorner \ll 1$) as

$$\begin{aligned} \theta^{Q'} &\leftarrow (1 - \urcorner)\theta^{Q'} + \urcorner\theta^Q, \\ \theta^{\mu'} &\leftarrow (1 - \urcorner)\theta^{\mu'} + \urcorner\theta^\mu. \end{aligned} \tag{26}$$

To ensure the exploration of the training samples, the action received from the actor network is added with the noise before acting on the environment. Then, at state $s(t)$, the obtained action is represented as

$$a(t) = \mu(s(t)|\theta^\mu) + \mathcal{OU}(t), \tag{27}$$

where $\mathcal{OU}(t)$ is chosen from the Ornstein-Uhlenbeck process.

As used in many off-policy reinforcement learning algorithms, DDPG also has a replay buffer for sampling experiences to update neural network parameters. After each transition, experiences including state, action, next state, and reward are combined as a tuple $(s(t), a(t), r(t), s(t+1))$, which is taken into the buffer. When training, the agent samples random mini-batches of experiences from the replay buffer to calculate and update the parameter of the main networks.

B. Proposed `FlyReflect` Framework

According to the above introduction, the `FlyReflect` framework is illustrated in Fig. 4, which is detailed as follows.

1) Framework Formulation: As the DDPG algorithm, the proposed framework consists of the environment, which is the whole system, the replay buffer with limited storage capacity, and four neural networks. We divide the framework into 2 parts: the interactive part (indicated with the red lines in Fig. 4) and learning part (the remaining part).

In the interactive part, from the environment, the state $s(t)$ is observed and the agent determines the action $a(t)$ based on the policy from the actor network acting on the environment, a mapping function is then used to adjudge the final action $a'(t)$ and act to the environment. Subsequently, the environment feeds back the next state $s(t+1)$ and the reward $r(t)$, and the tuple of experience is stored to the buffer for the training process.

In the learning part, a batch of experiences are randomly sampled from the buffer and the training process is performed based on the DDPG algorithm to update the networks.

The framework formulation is clearly defined as follows:

a) State space: As defined earlier, the position of the F-IRS at time slot t is represented as $p_I(t) = [x_I(t), y_I(t), z_I(t)]$, and the position of each GU can be denoted as $p_k(t) = [x_k(t), y_k(t), z_k = 0]$. Then, we rewrite the state (18) as

$$\begin{aligned} s(t) = \{&x_I(t), y_I(t), z_I(t), \\ &x_1(t), y_1(t), x_2(t), y_2(t), \dots, x_K(t), y_K(t)\}. \end{aligned} \tag{28}$$

The number of entries from the position of the F-IRS is 3, and that from the position of all GUs is $2K$. In summary, the state space has a dimension of $(3 + 2K)$, which is also the number of input nodes of the actor network.

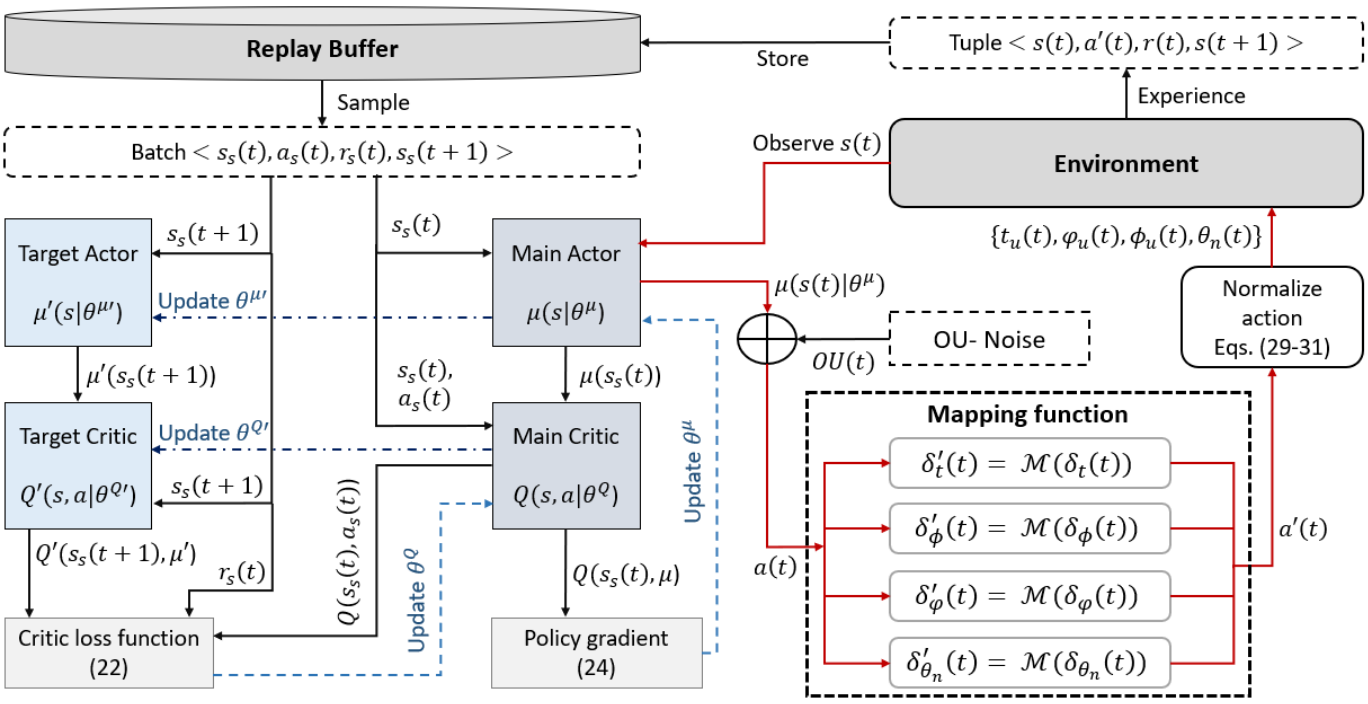


Fig. 4: Proposed `FlyReflect` framework.

b) Action space: As introduced in section III-D, the action at time slot t is defined as (19), in which the constraints in (17) need to be satisfied. Then, we normalize the action as follows

$$t_u(t) = \delta_t(t)\Delta_t, \quad \varphi_u(t) = \delta_\varphi(t)2\pi, \quad \phi_u(t) = \delta_\phi(t)\pi, \quad (29)$$

$$\theta_n(t) = \delta_{\theta n}(t)2\pi, \quad \forall n \in N, \quad (30)$$

where δ_t, δ_φ, δ_ϕ, $\delta_{\theta n} \in [0, 1]$. With the range from 0 to 1, the constraints in (17b), (17c), and (15) are satisfied. To meet the constraint (16), we normalize the elevation angle of the moving direction as

$$\phi_u(t) = \begin{cases} \pi/2 & \text{, if } \delta_\phi \pi \notin \left[\frac{z_{min} - z_I(t)}{v_c t_u(t)}, \frac{z_{max} - z_I(t)}{v_c t_u(t)}\right], \\ \delta_\phi(t)\pi & \text{, otherwise,} \end{cases} \quad (31)$$

where the F-IRS will fly horizontally ($\phi_u(t) = \pi/2$) if the action is out of the range in constraint (16). With the above normalization, all the constraints in (17) are satisfied. Then, the action space of the proposed framework is

$$a(t) = \{\delta_t(t), \delta_\phi(t), \delta_\varphi(t), \delta_{\theta 1}(t), \delta_{\theta 2}(t), \ldots, \delta_{\theta N}(t)\}. \quad (32)$$

Accordingly, the action space has a dimension of $(3 + N)$, which is also the number of output nodes of the actor network.

c) Mapping Function: Even though the actions are normalized in (29) and (30) to satisfy the constraints (17b), (17c), and (15), the values of the actions may be over the range $[0, 1]$ and violate the constraints due to the addition of noise for exploration in (27), in which the Ornstein-Uhlenbeck process is given by a differential equation as

$$d\mathcal{OU}(t) = \mho(\chi_o - \mathcal{OU}(t))dt + \varpi dW(t), \quad (33)$$

where $\mho$ is the rate of mean reversion, χ_o is the long-term mean of the process, ϖ is the average magnitude of the Wiener process $dW(t)$. The noise was considered in the entire $5e^6$ steps with $\mho = 0.2$, $\chi_o = 0$, $\varpi = 0.15$, and the Wiener process with mean $= 0$ and variance $= 1$. Probability density function (PDF) of the action values after adding the noise are illustrated in Fig. 5 as an example. In these examples, action values may go out of the range $[0, 1]$ to reach extension of $(-1, 2)$. Then, according to (29) and (30), the ranges of values of $t_u(t)$, $\varphi_u(t)$, $\phi_u(t)$, and θ_n are given by

$$\begin{aligned} -1 < t_u(t) < 2\Delta_t, \\ -\pi < \phi_u(t) < 2\pi, \\ -2\pi < \varphi_u(t), \theta_n < 4\pi, \end{aligned} \quad (34)$$

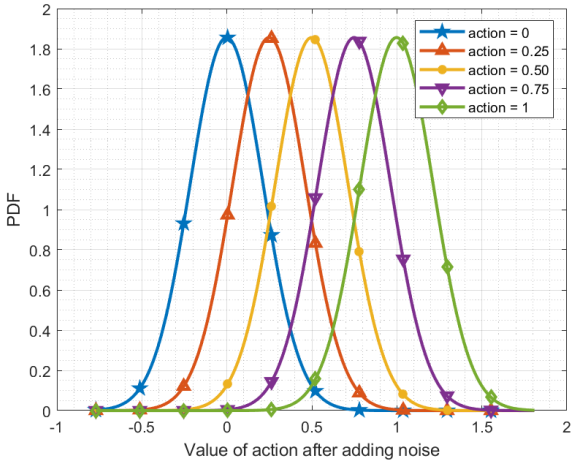


Fig. 5: Action after adding noise.

which violates the constraints (17b), (17c), and (15). As a result, these unrealistic action values lead to low accuracy and slow convergence during the training process.

To tackle the violation problem for the training process, we propose a mapping function to satisfy the range value of the actions in $[0, 1]$. The mapping function is defined as follows. For the action $\delta_t(t)$, which represents the moving time of the F-IRS in each time slot, we map it to the nearest suitable value when it is out of the range. The mapping function for $\delta_t(t)$ is written as

$$\delta_t'(t) = \begin{cases} 0 & , \text{if } \delta_t(t) < 0, \\ 1 & , \text{if } \delta_t(t) > 1, \\ \delta_t(t) & , \text{otherwise.} \end{cases} \tag{35}$$

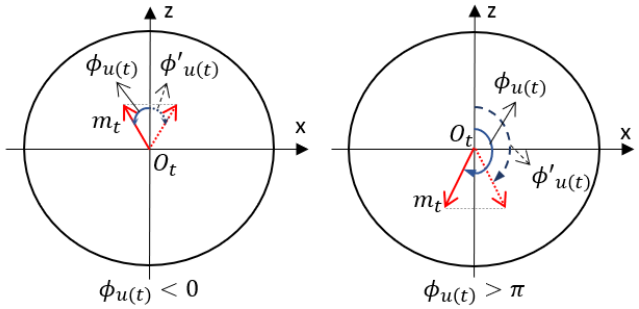
where $\delta_t' u(t)$ is the action after mapping.

Next, the value of $\phi_u(t)$ needs to be in the range [0,π]. This value represents how high the F-IRS moves in the time slot t; hence, we use a mapping function so that after mapping, the height of the movement does not change. We denote $\phi_u'(t)$ as the value after mapping, as illustrated in Fig. 6a with m_t is the vertical moving direction of the F-IRS, O_t is the altitude of the F-IRS, and the angle is measured clockwise starting from the z-axis. The value of $\phi_u'(t)$ is calculated as

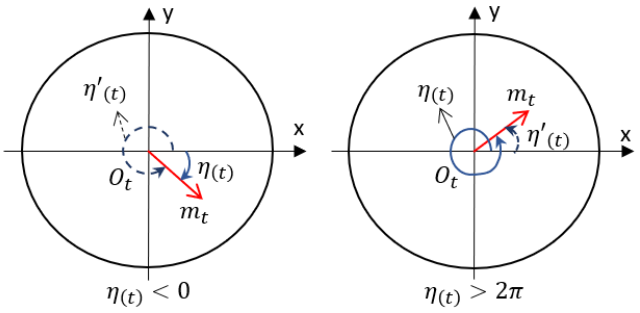
$$\phi_u'(t) = \begin{cases} -\phi_u(t) & , \text{if } \phi_u(t) < 0, \\ 2\pi - \phi_u(t) & , \text{if } \phi_u(t) > \pi. \end{cases} \tag{36}$$

Then, the action after mapping is

$$\delta_\phi'(t) = \begin{cases} -\delta_\phi(t) & , \text{if } \delta_t(t) < 0, \\ 2 - \delta_\phi(t) & , \text{if } \delta_t(t) > 1, \\ \delta_\phi(t) & , \text{otherwise.} \end{cases} \tag{37}$$



(a) $\phi_u(t)$ mapping.



(b) $\varphi_u(t), \theta_n(t)$ mapping.

Fig. 6: Action mapping function.

Algorithm 1: `FlyReflect` training algorithm

```
Set up the environment.
Initialize the main networks θ^μ, θ^Q by random.
Initialize the target networks θ^Q' ← θ^Q and θ^μ' ← θ^μ.
Initialize the model parameters and the experience replay buffer B
  with capacity C.
Set number of episodes E, number of steps T, batch size B
for e = 1,...,E do
    Get initial state s(t) of the episode as (28).
    while t < T do
        Interacting:
            Take action a(t) using (27).
            Mapping the action as (41): a'(t) = M(a(t)).
            Normalize action as in IV-B1b
            Perform a'(t) → get r(t), s(t+1).
            Accumulate episode reward.
            Store the tuple < s(t), a'(t), r(t), s(t+1) >.
            Update state s(t) ← s(t+1).
        Training:
            Sample the experiences from replay buffer.
            Update neural network parameters θ^Q, θ^μ
            Target networks soft update as (26).
    Calculate the average reward.
    if best average reward then
        Save the current main actor network: θ^μ* ← θ^μ
return the optimal networks θ^μ*.
```

For the remaining angle $\varphi_u(t)$ and phase shifts $\theta_n(t)$ with the range $[0, 2\pi]$, we denote an angle $\eta(t) \in [0, 2\pi]$ as the common value. Additionally, with the current position O_t, the action direction m_t and $\eta(t)$ is measured counter-clockwise starting from the x-axis, as illustrated in Fig. 6b, the value after mapping $\eta'(t)$ is defined such that the value on the x-y plane is not changed, as

$$\eta'(t) = \begin{cases} \eta_(t) + 2\pi & , \text{if } \eta(t) < 0, \\ \eta_(t) - 2\pi & , \text{if } \eta(t) > 2\pi. \end{cases} \tag{38}$$

Accordingly, the common action $\delta_\eta(t)$ for $\delta_\varphi(t)$ and $\delta_{\theta n}(t)$ after mapping is computed as

$$\delta_\eta'(t) = \begin{cases} \delta_\eta(t) + 1 & , \text{if } \delta_\eta(t) < 0, \\ \delta_\eta(t) - 1 & , \text{if } \delta_\eta(t) > 1, \\ \delta_\eta(t) & , \text{otherwise.} \end{cases} \tag{39}$$

In other words, the actions after mapping $\delta_\varphi'(t)$ and $\delta_{\theta n}'(t)$ are calculated as

$$\begin{aligned} \delta_\varphi'(t) &= \begin{cases} |\,|\delta_\varphi(t)| - 1\,| & , \text{if } \delta_\varphi(t) \notin [0,1], \\ \delta_\varphi(t) & , \text{otherwise.} \end{cases} \\ \delta_{\theta n}'(t) &= \begin{cases} |\,|\delta_{\theta n}(t)| - 1\,| & , \text{if } \delta_{\theta n}(t) \notin [0,1], \\ \delta_{\theta n}(t) & , \text{otherwise.} \end{cases} \end{aligned} \tag{40}$$

In summary, after mapping, the obtained action of the DRL framework is

$$a'(t) = \mathcal{M}(a(t)) = \{\delta_t'(t), \delta_\phi'(t), \delta_\varphi'(t), \delta_{\theta 1}'(t), \cdots, , \delta_{\theta N}'(t)\}, \tag{41}$$

where $\mathcal{M}(\cdot)$ is the mapping function as defined above.

2) Working Procedure: The process can be described as follows. At the initialization stage, the main network parameters, θ^μ and θ^Q, are randomly initialized and the target network

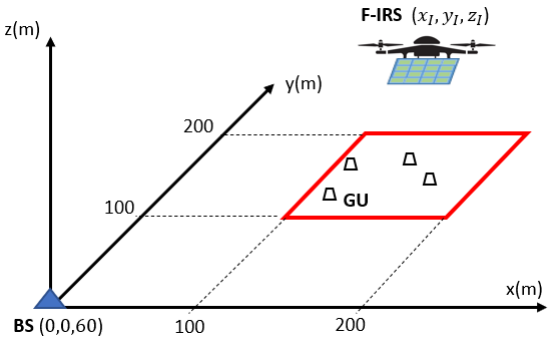


Fig. 7: Simulation scenario.

parameters, $\theta^{\mu'}$ and $\theta^{Q'}$, are directly copied form the main networks. Additionally, the experience replay buffer $\mathcal{B}$ with capacity $\mathcal{C}$ is built.

When interacting with the environment, at the first step of each episode, an initial state $s(t)$ is observed from the environment. At each step, it executes the action $a(t)$ using (27), the mapping function is then used to adjudge the action $a(t)$ to action $a'(t)$ as per (41). After that, the action is normalized as defined in IV-B1b and performs to the environment. Next, the reward $r(t)$ is received and the environment updates to the next state $s(t+1)$. Accordingly, the tuple of experiences is taken into the replay buffer, and the state $s(t)$ is updated by $s(t+1)$.

In the training process, the networks are trained according to the sampled experiences from the replay buffer. The main critic network parameter θ^Q is updated by minimizing the critic loss function calculated using (22). Then, the policy gradient as per (24) is executed to update the main actor network parameter θ^μ. The target networks $(\theta^{\mu'}, \theta^{Q'})$ are then updated according to (26). Finally, the optimal actor network is selected by comparing the average reward after each episode, and the network with the best average reward is returned after training for testing. The detailed algorithm is shown in Algorithm 1.

V. Performance Evaluation

A. Simulation Settings

We generate an environment to simulate and train the agent using Pytorch with Python programming for the evaluation. The hidden layers in this simulation use the rectified linear unit (ReLU) function as the activation function. The output of the critic network is obtained linearly, and the output layer of the actor network uses the Sigmoid activation function.

We consider a scenario including a BS with a height of 60 m that serves the GUs in an underserved area with a fixed range assisted by an F-IRS, the simulation environment is illustrated as Fig. 7. The initialized position of the F-IRS is fixed at (100,100,150) (m), and it moves at a velocity of $v_c = 15$ (m/s). The initialized positions of the GUs are uniformly distributed in the serving area, and each GU can move over the time with a moving probability from 0.05 to 0.15 per time step, which means the GU can move approximately 5 to 15 milliseconds in a period of 100 milliseconds. The directions of the movement are randomly sampled from four values of $0, \pi/2, \pi, 3\pi/2$.

TABLE II: Simulation parameters

Parameter		**Value**
Critic network	Hidden layer 1	512 nodes
	Hidden layer 2	1024 nodes
Actor network	Hidden layer 1	512 nodes
	Hidden layer 2	256 nodes
Target network update rate, τ		$1e^{-3}$
Number of training steps		500 steps
Time step, Δt		0.1 (s)
Height of the BS		60 (m)
Initialized position of the F-IRS		(100,100,150) (m)
Velocity of the UAV, v_c		15 (m/s)
Movement of the users	Probability	5-15 %
	Direction	$\{0, \pi/2, \pi, 3\pi/2\}$
Path loss exponents, β_1, β_{2k}		2.8
Reference path loss, C_0		-30dB
Rician factors, ε_1, ε_{2k}		∞
Transmit power, P_T		10dBm
Noise power, σ^2		-170dBm/Hz

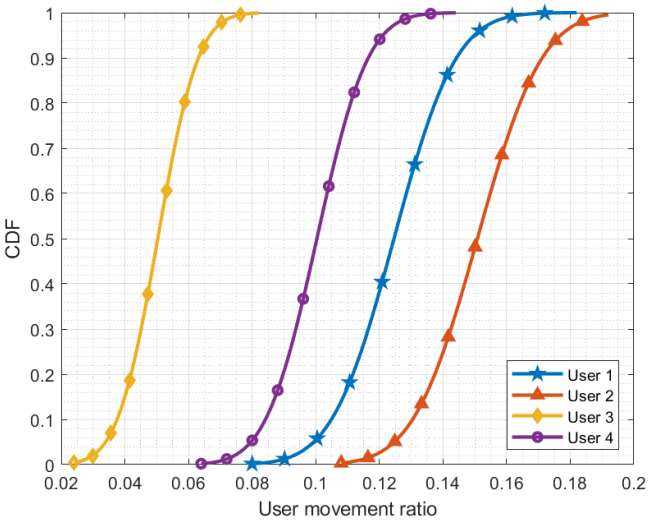


Fig. 8: CDF of the movement ratio of the GUs.

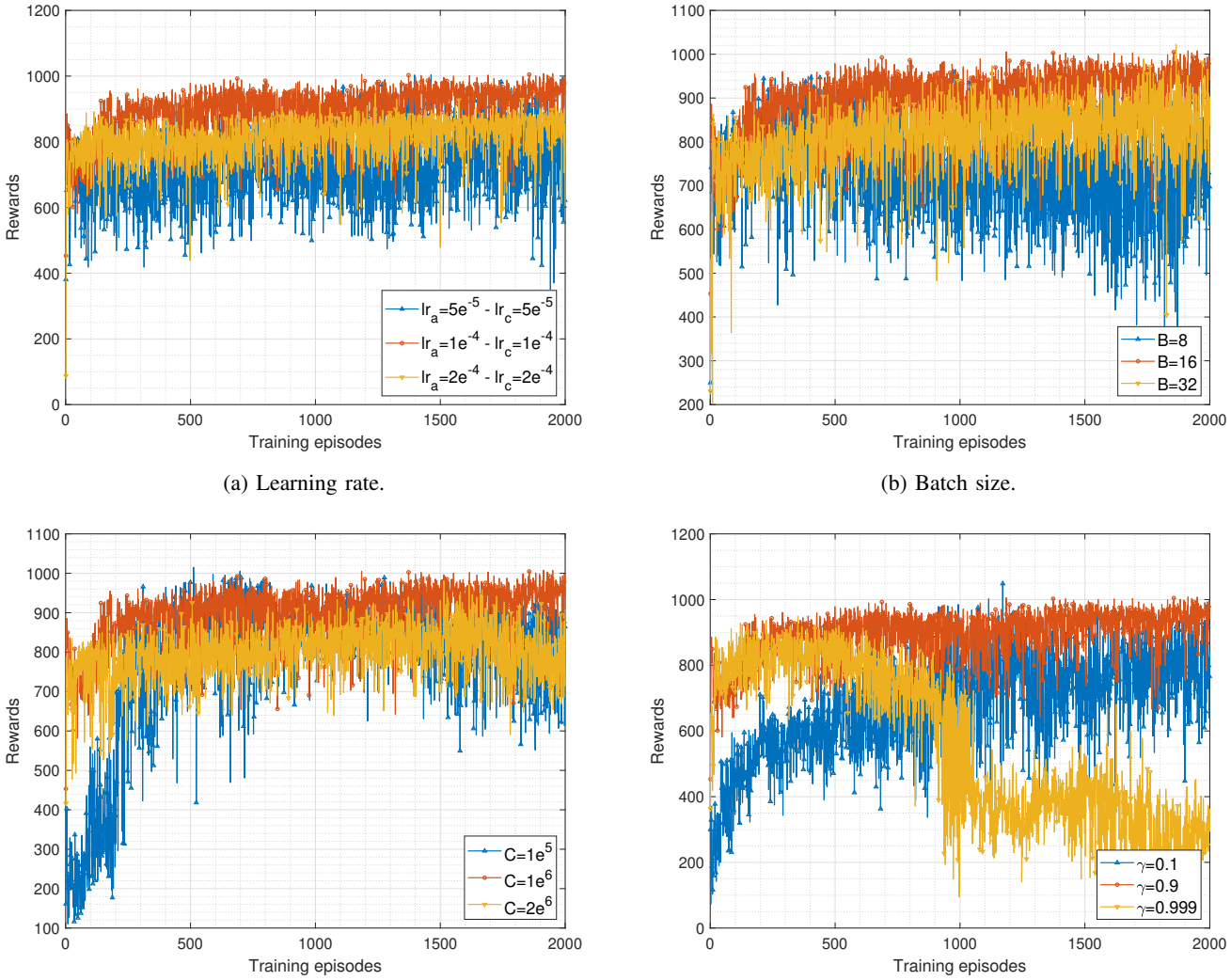
We calculate the path losses for the channels according to the distance-dependent path loss model [32], it is calculated as

$$PL(d) = C_0 \times (\frac{d}{D_0})^{\beta}, \tag{42}$$

where β, d, and C_0 are the path loss exponent, channel distance, and reference path loss at the reference distance D_0, respectively. Accordingly, the corresponding path losses of the B2F link and the F2k link are calculated as

$$\begin{aligned} L_1 &= C_0 \times (\frac{d_{B2F}}{D_0})^{\beta_1}, \\ L_{2k} &= C_0 \times (\frac{d_{F2k}}{D_0})^{\beta_{2k}}, \end{aligned} \tag{43}$$

where β_1 and β_{2k} are the corresponding path loss exponents. In this simulation, we set $\beta_1 = \beta_{2k}$ =2.8, and $C_0 = -30$ dB. The antenna array at the BS in this simulation has 6 elements



(a) Learning rate.

(b) Batch size.

(c) Buffer capacity.

(d) Discount factor.

Fig. 9: Monte-Carlo experiments for training hyperparameters.

with 2 on the vertical and 3 on the horizontal. The spacing between the array elements of the BS's antenna and the IRS elements are chosen with the same value as $d_u = d_v = d_p = d_q = \frac{\lambda}{2}$. The transmit beamforming vectors $\mathbf{w_k}$ in this study are randomly generated for each simulation case, which satisfy power constraint as

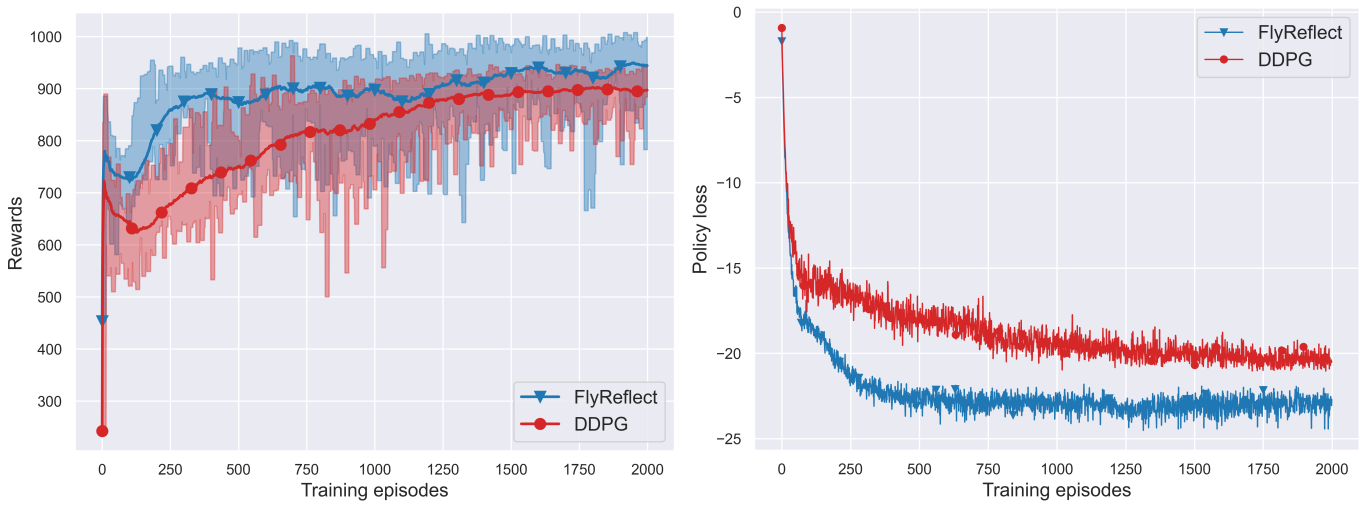
$$\sum_{w=1}^{K} ||\mathbf{w_k}||^2 = P_T, \tag{44}$$

where P_T is the transmit power at the BS. We summarize the parameters of this simulation in Table II.

Before evaluating the performance, we estimate the dynamic of the simulation environment using movement ratio. The movement ratio is the ratio that the GU moves during the entire considered period time. At the beginning of the simulation, we set the probability values for the movement of the GUs, which are randomly chosen in the range of 5–15%. Then, we test the environment with 4 GUs in 500 episodes, each episode has 500 time steps. The obtained movement ratios in each episode of the GUs are expressed by the cumulative distribution function (CDF) as illustrated in Fig. 8. The GU 2 moves the most when the movement ratio is from 10.8% to 19.15%, whereas the GU 3 is the least with the ratio of 2.4–8.2%. Besides, we calculate the mean of the movement ratios over 500 episodes and the mean ratios of the GU 1 to GU 4 are 12.46%, 15.08%, 5.03%, and 10.04%, respectively. This implies that the dynamics of the GUs in the simulation environment under testing match the expectation when the movement probabilities are set from 5% to 15%.

B. Convergence Analysis

Finding appropriate hyperparameters for the model is a hurdle in deep reinforcement algorithms. To do so, we conduct a Monte-Carlo experiment to reveal the model's best-fit hyperparameters. We simulate a scenario with four GUs and a 16-element F-IRS (size 4x4) in this sub-section to see how the hyperparameters affect convergence in training the DRL model.



(a) Training rewards.

(b) Policy loss.

Fig. 10: Convergence comparison between proposed `FlyReflect` framework and DDPG scheme.

At first, we turn the learning rate of the model by selecting three cases of actor learning rate (lr_a) and critic learning rate (lr_c) to experiment and get the results as in Fig. 9a. The results show that the rewards converge to specific ranges in two cases of $(\text{lr}_a, \text{lr}_c) = \{(1e^{-4}, 1e^{-4}), (2e^{-4}, 2e^{-4})\}$, while it still varies a lot after 2000 episodes in the remaining case. Because the too small learning rates result in too small update gradient steps, which is hard to meet the optimal point. Furthermore, with a high learning rate, the model may oscillate around but cannot move into the optimal. Therefore, case $(\text{lr}_a, \text{lr}_c) = (1e^{-4}, 1e^{-4})$ gives the best performance, where the average reward is approximate 10.04% and 23.98% higher than the case of $(2e^{-4}, 2e^{-4})$ and $(1e^{-5}, 1e^{-5})$, respectively.

Next, we experiment to see how the batch size B affects the results. It describes the number of samples utilized in each gradient update step, which affects the model's learning speed and stability. Using fewer samples leads to a less accurate estimate of the error gradient due to the heavy reliance on the specific training samples. In a dynamic environment, on the other hand, a high batch size can produce noise due to the changing state of the environment and may cause poor convergence, it also requires more resources for computation. For evaluating the impact of batch size in the considered model, we try training the model with three different batch sizes ranging from 8 to 32, and the results are in Fig. 9b. The results show that when the batch size has the value of 16, the model shows the best performance. In the other cases, the convergence is worse due to the noise when the batch size is large (32) and cannot converge due to the bias when the batch size is small (8).

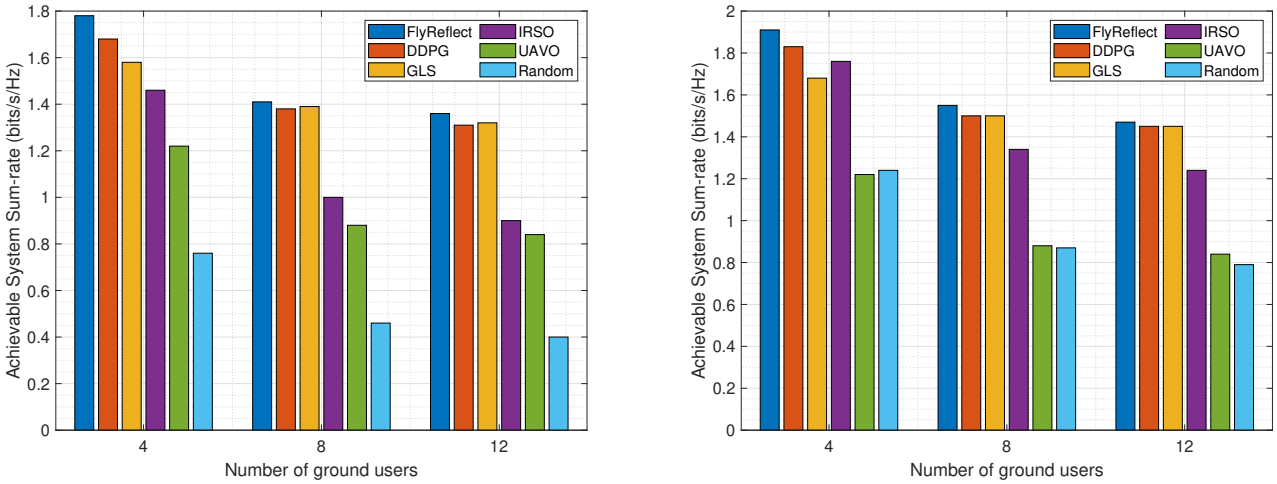
Besides, we experiment with the impact of the replay buffer size on convergence. In this experiment, we choose three values of the buffer size C as $\{1e^5, 1e^6, 2e^6\}$ to observe the performance, and the results are illustrated in Fig. 9c. If the number of experiences exceeds the capacity of the replay buffer, the oldest experiences will be replaced by incoming experiences. As a result, due to the small size of the buffer, some experiences may not be used to train the model, and convergence may suffer the effects of inadequate exploration samples. As a consequence, due to a lack of exploration sample, the reward with the buffer size $1e^5$ does not improve after 800 episodes. Furthermore, the reward in the case $C = 2e^6$ begins to decline after 1700 episodes. Because of the large buffer size, the experience data contains a lot of old data, which increases the risk of sampling old data, then detrimentally affect the learning process. Therefore, $C = 1e^6$ is suitable for this model, which gives the best training result.

Finally, we experiment to observe the impact of the discount factor γ. The discount factor decides the attention of the agent to the future. If γ =0, the agent only learns based on an immediate reward, and when γ =1, the agent cares about the sum of all future rewards. In this experiment, we choose threes values of the discount factor {0.1, 0.9, 0.999} to perform the training. The rewards in Fig. 9d show that if the agent only considers the immediate and very near future rewards in a dynamic environment, the reward cannot effectively reflect the quality of the action to the entire environment for a long time with the changes, the model was difficult to convert with a small discount factor (0.1). Similarly, due to the big difference from the essence of the environment, a significant big discount factor (0.999) makes convergence impossible. In this environment, $\gamma = 0.9$ is a good fit and provides the best result.

C. Performance of the Proposed Framework

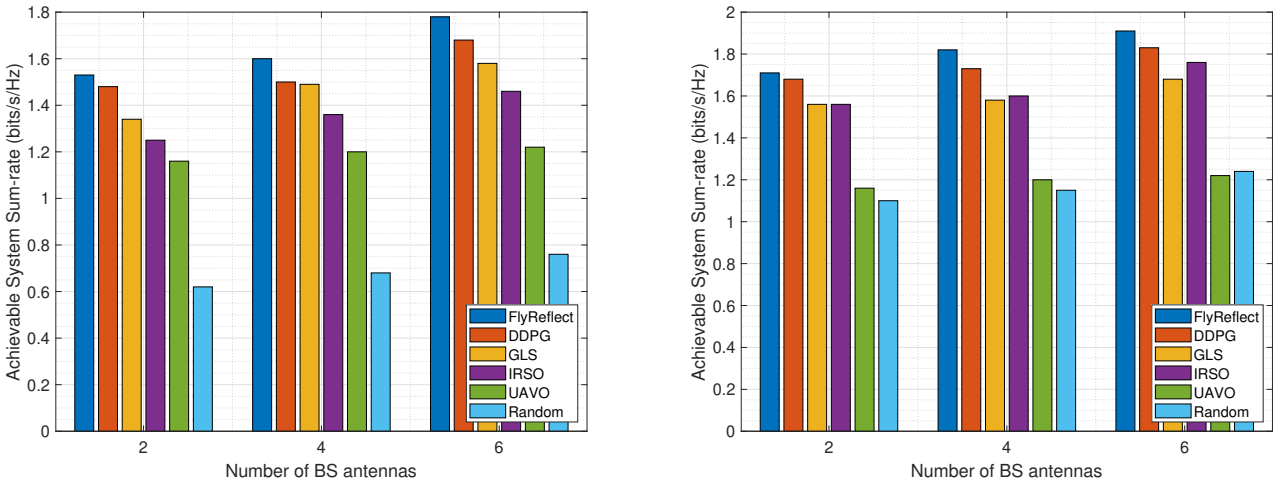
To demonstrate the efficiency of the proposed framework, we compare the performance of the proposed framework with some benchmark schemes, which are defined as follows.

- *Original DDPG* (DDPG): This is the scheme that trains the action using the DDPG algorithm as in [33], in which the action after adding noise is used to interact with the environment without the mapping function. To deal with the action violation as analyzed in IV-B1c, we directly trim the action to the range of $[0, 1]$.



(a) IRS size: 2x2. (b) IRS size: 4x4.

Fig. 11: Achievable system sum-rate concerning the number of ground users with different sizes of IRS.



(a) IRS size: 2x2. (b) IRS size: 4x4.

Fig. 12: Achievable system sum-rate concerning the number of BS antennas with different sizes of IRS.

- *Greedy using local searching* (GLS): In this approach, we quantize continuous actions into discrete space and find the optimum action at each time step. In a vast action space, however, examining all conceivable actions in discrete space is unfeasible. Then, we applied a local search with low complexity as presented in [41] to pick the action that delivers the best reward at each time step.
- *IRS only* (IRSO): This scheme is used to make a comparison to evaluate the improvement of the F-IRS model. As in the traditional IRS that is built on a fixed physical structure, we fixed the position of the airborne vehicle, and only optimize the IRS phase shift matrix. This is also the F-IRS model but at a lower flexible level, it is nearly the fixed IRS scheme.
- *Airborne vehicle only* (UAVO): In contrast with IRSO, this scheme assumes a communication network independent of the IRS assistance network. In this scheme, the airborne vehicle movement is optimized according to changes in the environment, while the phase shift matrix is set to one element with a value of zero.
- *Random*: The actions in this scheme are picked at random from the constraints range for each time slot.

Firstly, the exploration of the proposed framework is compared with the original DDPG algorithm. We train the model with the same parameters for both the proposed and the DDPG algorithm and get the results after 2000 episodes in Fig. 10. We illustrate the training rewards in Fig.10a with the fuzzy part is the episode reward and the other part is the average reward. As a result, the proposed framework outperforms the DDPG scheme in terms of exploration, where the maximum rewards are 1018.78 and 963.4, respectively. Because trimming the action value directly to the range [0,1] in DDPG restricts exploring new samples while using the mapping function as the proposed framework ensures the exploration for training.

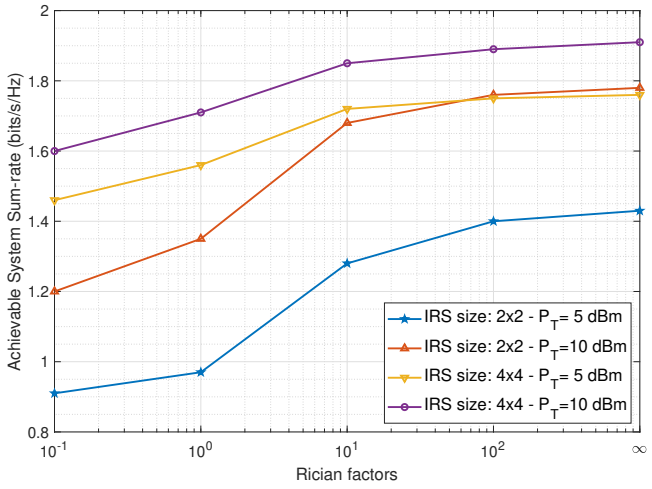


Fig. 13: Performance of `FlyReflect` framework in fading environment

In addition, we evaluate the policy loss of two schemes in Fig.10b. The policy loss is used to estimate the quality of the actor network, which is calculated as

$$\mathcal{L} = -\frac{1}{B}\sum_{b=1}^{B} Q\left(s_b, \mu(s_b|\theta^{\mu})|\theta^{Q}\right), \tag{45}$$

where $Q\left(s_b, \mu(s_b|\theta^{\mu})|\theta^{Q}\right)$ is the action-value function of the policy, which implies that the higher the value of Q, the smaller the value of the loss function, which means the better the agent network. The results show that the policy loss of the proposed `FlyReflect` framework converges after about 1250 episodes while it is about 1750 episodes in DDPG. In addition, `FlyReflect` gives approximately 20.3% better than the DDPG.

Next, we experiment to observe the effects of the number of GUs (K) on system performance with different sizes of IRS. We simulate a scenario with 6-antenna BS in two cases of F-IRS size, which are 4 elements (size 2x2) and 16 elements (size 4x4). The obtained results in Fig.11 show that the ASSR decreases when K increases. Because the number of GUs increases, the interference signals from other GUs grow, causing a decrease in SINR and hence a reduction in the achievable rate. Besides, `FlyReflect` gives the best performance in all cases, where it is approximately 4.12%, 6.06%, 35.42%, 54.76%, and over 100% better than DDPG, GLS, IRSO, UAVO, and Random, respectively. In addition, with a large size of F-IRS, the system can improve the achieved result, where the 16-element F-IRS gives around 7.4% better than the 4-element F-IRS performance. Furthermore, a large size of F-IRS also discrete the ASSR reduction when increasing the GUs. Specifically, when expanding from 4 to 8 GUs, the ASSR reduces 18.8% and 20.8% in 16-element and 4-element F-IRS, respectively. Therefore, in practice, a trade-off between the scale of the environment and the size of IRS needs to be considered.

Then, for observing the system performance when changing the number of BS antennas, we model a scenario with 4 GUs and two sizes of F-IRS and get results in Fig. 12. The ASSR are increased when increasing the value of M, because with the higher diversity of communication spatial, the easier it is for an agent to choose the action to increase the performance. The result also shows that our `FlyReflect` performs better than the others in all the cases of M, where it is about 5.36%, 11.34%, 20.64%, 37.15%, and 138.3% better than DDPG, GLS, IRSO, UAVO, and Random, respectively. Besides, the case of 16-element F-IRS gives approximately 10.79% higher ASSR than the 4-element F-IRS case. Again, this demonstrates that the large size of F-IRS enhances the system performance better than the small size of F-IRS. Furthermore, in the case of the 16-element IRS, IRSO has higher results than GLS owing to two reasons. Firstly, the discrete actions with local search may get stuck at a poor optimum point in vast action space. And second, large size IRS can significantly improve performance in a diverse spatial environment.

Finally, we evaluate the proposed framework performance in fading channels with different transmit powers. The fading channels are applicable models for simulating real-world phenomena such as multi-path scattering, temporal dispersion, and Doppler shifts caused by relative motion between the transmitter and receiver in wireless communications. We observe the ASSR in an environment with $K = 4$, $M = 6$, and the variation of the Rician factors. The results are illustrated in Fig.13 with five values of the Rician factors as $\varepsilon_1 = \varepsilon_{2k} = \{10^{-1}, 10^{0}, 10^{1}, 10^{2}, \infty\}$ and two values of transmit power $P_T = \{5, 10\}$ (dBm). As a result, increasing the transmit power improves the achieved ASSR, where $P_T = 10$ (dBm) gains 8.52% and 24.48% compared with $P_T = 5$ (dBm) in the cases of 16-element and 4-element F-IRS, respectively. Furthermore, the ASSR increases in tandem with the value of the Rician factor and otherwise. Because the smaller the Rician factor value, the higher the NLoS component in the communication channel (3), which leads to a chaotic signal for the communication channel and hence reduces the system performance. Specifically, the Rician factors are ∞ demonstrating for the LoS channel, where there is no chaotic signal, and it gives the best performance with approximate 1.18%, 5.36%, 23.08%, and 33.08% higher than 10^2, 10^1, 10^0, and 10^{-1}, respectively. We can observe that the proposed framework can perform well in a slight fading environment, with the result in case 10^2 being close to the ideal scenario (∞). Consequently, a fading environment with a high chaotic signal might harm system performance. To deal with this problem, the agent can monitor and predict the chaotic components by expanding the state space with the NLoS components, allowing it to decide a suitable action at each time slot.

VI. Conclusion

In this paper, we have considered a downlink multi-user MISO F-IRS system model. The communication channel models have been analyzed, wherein the changes of the channel gains depend on the position of the F-IRS and the phase shift matrix. Accordingly, we formulated an optimization problem that maximizes the ASSR by joint optimizing the flying trajectory and the phase shift matrix of the IRS. To solve the problem, we have transformed it into an MDP-based problem and proposed a DRL framework based on the

DDPG algorithm with the addition of a mapping function to improve the effectiveness of the DDPG algorithm, referred to as `FlyReflect`. The experiment results show that our proposed framework significantly improves the convergence where it improved the policy loss approximately 20.3% better than the DDPG algorithm. In addition, the simulation results exposed the outperformance of the F-IRS model compared with benchmark approaches. In particular, `FlyReflect` improves the ASSR approximately 34.52% and 54.76% greater than the IRSO and UAVO approach, respective. On the other hand, the simulation illustrates the considerable effects of the number of GUs and the number of BS antennas on the system, in which, an increase of the number of GUs results in a decrease in the ASSR owing to the interference. To tackle this issue, the NOMA technology could be considered in future work for a dense multi-user environment. In addition, optimization of the transmit beamforming matrix from BS to F-IRS is also one of the interesting research direction as well as cell-free MIMO system, uplink transmission, and multiple F-IRS cooperation.

ACKNOWLEDGMENT

This research was supported by the Chung-Ang University Young Scientist Scholarship in 2020.

REFERENCES

[1] L. Chettri and R. Bera, "A comprehensive survey on Internet of things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[3] B. Matthiesen, E. Björnson, E. De Carvalho, and P. Popovski, "Intelligent reflecting surface operation under predictable receiver mobility: A continuous time propagation model," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 216–220, 2021.

[4] Z. Peng, Z. Zhang, C. Pan, L. Li, and A. L. Swindlehurst, "Multiuser full-duplex two-way communications via intelligent reflecting surface," *IEEE Transactions on Signal Processing*, vol. 69, pp. 837–851, 2021.

[5] S. Sugiura, Y. Kawai, T. Matsui, T. Lee, and H. Iizuka, "Joint beam and polarization forming of intelligent reflecting surfaces for wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1648–1657, 2021.

[6] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, "Robust and secure wireless communications via intelligent reflecting surfaces," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2637–2652, 2020.

[7] P. Mursia, V. Sciancalepore, A. Garcia-Saavedra, L. Cottatellucci, X. C. Pérez, and D. Gesbert, "RISMA: Reconfigurable intelligent surfaces enabling beamforming for IoT massive access," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1072–1085, 2021.

[8] C. Pradhan, A. Li, L. Song, J. Li, B. Vucetic, and Y. Li, "Reconfigurable intelligent surface (RIS)-enhanced two-way OFDM communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16270–16275, 2020.

[9] I. Yildirim, A. Uyrus, and E. Basar, "Modeling and analysis of reconfigurable intelligent surfaces for indoor and outdoor applications in future wireless networks," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1290–1301, 2021.

[10] D. Dardari, "Communicating with large intelligent surfaces: Fundamental limits and models," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2526–2537, 2020.

[11] W. Yan, X. Yuan, and X. Kuai, "Passive beamforming and information transfer via large intelligent surface," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 533–537, 2020.

[12] R. C. Ferreira, M. S. P. Facina, F. A. P. De Figueiredo, G. Fraidenraich, and E. R. De Lima, "Bit error probability for large intelligent surfaces under double-nakagami fading channels," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 750–759, 2020.

[13] S. Gong, X. Lu, D. T. Hoang, D. Niyato, L. Shu, D. I. Kim, and Y.-C. Liang, "Toward smart wireless communications via intelligent reflecting surfaces: A contemporary survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2283–2314, 2020.

[14] N.-N. Dao, Q.-V. Pham, N. H. Tu, T. T. Thanh, V. N. Q. Bao, D. S. Lakew, and S. Cho, "Survey on aerial radio access networks: Toward a comprehensive 6G access infrastructure," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 1193–1225, 2021.

[15] B. Di, H. Zhang, L. Song, Y. Li, Z. Han, and H. V. Poor, "Hybrid beamforming for reconfigurable intelligent surface based multi-user communications: Achievable rates with limited discrete phase shifts," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1809–1822, 2020.

[16] M. Fu, Y. Zhou, and Y. Shi, "Intelligent reflecting surface for downlink non-orthogonal multiple access networks," in *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2019.

[17] X. Mu, Y. Liu, L. Guo, J. Lin, and N. Al-Dhahir, "Exploiting intelligent reflecting surfaces in NOMA networks: Joint beamforming optimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6884–6898, 2020.

[18] Q. Wang, F. Zhou, R. Q. Hu, and Y. Qian, "Energy efficient robust beamforming and cooperative jamming design for IRS-assisted MISO networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2592–2607, 2021.

[19] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[20] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2042–2055, 2021.

[21] L. Yang, F. Meng, J. Zhang, M. O. Hasna, and M. D. Renzo, "On the performance of RIS-assisted dual-hop UAV communication systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10385–10390, 2020.

[22] Y. Pan, K. Wang, C. Pan, H. Zhu, and J. Wang, "UAV-assisted and intelligent reflecting surfaces-supported terahertz communications," *IEEE Wireless Communications Letters*, vol. 10, no. 6, pp. 1256–1260, 2021.

[23] H. Lu, Y. Zeng, S. Jin, and R. Zhang, "Aerial intelligent reflecting surface: Joint placement and passive beamforming design with 3D beam flattening," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[24] T. Shafique, H. Tabassum, and E. Hossain, "Optimization of wireless relaying with flexible UAV-borne reflecting surfaces," *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 309–325, 2021.

[25] T. Zhou, K. Xu, X. Xia, W. Xie, and J. Xu, "Achievable rate optimization for aerial intelligent reflecting surface-aided cell-free massive MIMO system," *IEEE Access*, vol. 9, pp. 3828–3837, 2021.

[26] M. Samir, M. Elhattab, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Optimizing age of information through aerial reconfigurable intelligent surfaces: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3978–3983, 2021.

[27] H. Yu, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Joint design of reconfigurable intelligent surfaces and transmit beamforming under proper and improper gaussian signaling," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2589–2603, 2020.

[28] H. Guo, Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3064–3076, 2020.

[29] H. Liu, X. Yuan, and Y.-J. A. Zhang, "Matrix-calibration-based cascaded channel estimation for reconfigurable intelligent surface assisted multiuser MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2621–2636, 2020.

[30] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1499–1513, 2014.

[31] S. K. Yong and J. Thompson, "Three-dimensional spatial fading correlation models for compact MIMO receivers," *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 2856–2869, 2005.

[32] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[33] K. Feng, Q. Wang, X. Li, and C.-K. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 745–749, 2020.

[34] C. Huang, G. Chen, and K.-K. Wong, "Multi-agent reinforcement learning-based buffer-aided relay selection in IRS-assisted secure cooperative networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4101–4112, 2021.

[35] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[36] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2021.

[37] H. Yang, Z. Xiong, J. Zhao, D. Niyato, Q. Wu, H. V. Poor, and M. Tornatore, "Intelligent reflecting surface assisted anti-jamming communications: A fast reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1963–1974, 2021.

[38] C. Huang, G. Chen, Y. Gong, M. Wen, and J. A. Chambers, "Deep reinforcement learning-based relay selection in intelligent reflecting surface assisted cooperative networks," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1036–1040, 2021.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* The MIT Press, 2018.

[40] T. P. Lillicrap, J. J. Hunt, A. e. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv e-prints*, p. arXiv:1509.02971, Sept. 2015.

[41] X. Ma, Z. Chen, W. Chen, Z. Li, Y. Chi, C. Han, and S. Li, "Joint channel estimation and data rate maximization for intelligent reflecting surface assisted terahertz MIMO communication systems," *IEEE Access*, vol. 8, pp. 99565–99581, 2020.