Reinforcement Learning-Based Downlink Transmit Precoding for Mitigating the Impact of Delayed CSI in Satellite Systems

Yasaman Omid *Student Member, IEEE*, Marios Aristodemou *Student Member, IEEE*, Sangarapillai Lambotharan *Senior Member, IEEE*, Mahsa Derakhshani *Senior Member, IEEE*, Lajos Hanzo *Life Fellow, IEEE*

***Abstract*—The integration of low earth orbit (LEO) satellites with terrestrial communication networks holds the promise of seamless global connectivity. The efficiency of this connection, however, depends on the availability of reliable channel state information (CSI). Due to the large space-ground propagation delays, the estimated CSI is outdated. In this paper we consider the downlink of a satellite operating as a base station in support of multiple mobile users. The estimated outdated CSI is used at the satellite side to design a transmit precoding (TPC) matrix for the downlink. We propose a deep reinforcement learning (DRL)-based approach to optimize the TPC matrices, with the goal of maximizing the achievable data rate. We utilize the deep deterministic policy gradient (DDPG) algorithm to handle the continuous action space, and we employ state augmentation techniques to deal with the delayed observations and rewards. We show that the DRL agent is capable of exploiting the time-domain correlations of the channels for constructing accurate TPC matrices. This is because the proposed method is capable of compensating for the effects of delayed CSI in different frequency bands. Furthermore, we study the effect of handovers in the system, and show that the DRL agent is capable of promptly adapting to the environment when a handover occurs.**

***Index Terms*—LEO Satellite Communication, Non-terrestrial Networks, 6G, Delayed CSI, Machine Learning, Reinforcement Learning**

I. Introduction

THE integration of satellite networks with terrestrial communication systems holds the potential of supporting seamless global connectivity [1]–[3]. In particular, low Earth orbit (LEO) satellites, which operate at altitudes between 300 km and 2000 km, can provide extended coverage to numerous devices simultaneously, while maintaining a relatively low latency of 1–7 msec [4]. Additionally, LEO mega-constellations, such as Starlink, offer widespread access to a vast network of satellites [5], and with the inevitable escalation of user population and data traffic, these constellations can be harnessed to build a global network supporting numerous devices [6], [7]. Although unmanned aerial vehicles (UAVs) have been widely considered as supplements to terrestrial communication networks, they can only provide short-term connectivity, and their coverage is significantly more limited compared to satellites. Therefore, exploring the integration of non-terrestrial networks with terrestrial communication systems is a compelling objective for future systems [8]–[11].

The integration of LEO satellite constellations with terrestrial networks presents significant challenges, particularly in providing high-throughput connectivity to users. Some of these challenges are high Doppler shifts, frequent handovers, and the propagation delays due to the considerable distance between users and satellites [12]. This propagation delay often exceeds the channel's coherence interval, causing difficulties in the channel estimation process in the uplink. In traditional pilot-based channel estimation techniques using time-division duplexing (TDD), users transmit unique pilot sequences, allowing the satellite to estimate the channels between itself and the users. However, the long propagation delay leads to a mismatch between the estimated channel state information (CSI) used for transmit preprocessing and the actual channel conditions experienced during downlink transmission, a challenge known as the "outdated CSI problem" [13]. This issue is a critical obstacle in the downlink of satellite networks and it is not limited to TDD systems—it also affects frequency-division duplexing (FDD) systems. Addressing this challenge is essential for improving the efficiency and reliability of satellite communication systems.

The downlink signal transmission from a multi-antenna satellite to multiple users is widely studied in the literature. For instance, in [14] and [15], downlink transmission protocols

Yasaman Omid, Marios Aristodemou and Mahsa Derakhshani are with the Wolfson School of Mechanical Electrical and Manufacturing Engineering at Loughborough University, Loughborough, U.K. (e-mail: y.omid@lboro.ac.uk; M.Aristodemou@lboro.ac.uk; M.Derakhshani@lboro.ac.uk).

Sangarapillai Lambotharan is with the Institute for Digital Technologies, Loughborough University London, London, UK (email: s.lambotharan@lboro.ac.uk)

Lojas Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton, UK (email: lh@ecs.soton.ac.uk)

M. Derakhshani and S. Lambotharan would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council (EPSRC) projects under grant EP/X012301/1, EP/X04047X/1, and EP/Y037243/1. L. Hanzo would like to acknowledge the financial support of EPSRC projects under grant EP/Y026721/1, EP/W032635/1 and EP/X04047X/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council (EPSRC) projects under grant EP/Y026721/1, EP/W032635/1 and EP/X04047X/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028).

were developed for a satellite equipped with a uniform planar array (UPA). These protocols supported communication with either multiple single-antenna users [14] or multiple user terminals equipped with UPAs [15]. Both studies focused on utilizing statistical CSI in their downlink transmit precoding (TPC) designs, thereby avoiding the complexities associated with instantaneous CSI. Also, in [16], the downlink transmission of multiple LEO satellites equipped with UPAs to several ground users is considered, and to avoid the outdated CSI problem, the authors used the average CSI to design downlink beamforming vectors. However, upon relying solely on statistical or average CSI and disregarding the short-term variations of the channel, there is a significant potential throughput erosion. This is particularly problematic for mobile handheld receivers.

The consideration of instantaneous CSI imposes the challenge of the outdated CSI, a topic that has been studied in several research papers.

The authors of [13] proposed a deep learning (DL) approach using long short-term memory units to predict satellite channel conditions and mitigate the effects of outdated CSI. Unlike previous solutions that either compensated for performance degradation or assumed perfect CSI [17], their method employed deep neural networks (DNN) to predict channel conditions, effectively addressing the challenge of outdated CSI. In [6], the authors proposed a pair of DNN systems, one for satellite channel prediction (SatCP) and another for satellite hybrid beamforming (SatHB). In SatCP, by exploiting the correlation between the uplink and the downlink CSI, a DNN was employed for predicting the downlink CSI in both TDD and FDD-based scenarios. Then, the output of the SatCP network was fed into the SatHB network. The SatHB network then harnessed the estimated CSI for constructing a hybrid beamformer. The literature includes a large body of research dedicated to channel prediction and the problem of channel ageing in terrestrial communications [18]–[21]. However, dealing with this problem in non-terrestrial communications with much longer delays needs further investigation. Further advances on dealing with outdated CSI in satellite systems have also been discussed in [22]–[25].

Against the above backdrop, in this paper we consider the downlink transmission of a satellite in support of multiple users on the ground. Here, we adopt the TDD technique due to its lower CSI overhead and because it is the preferred mode of operation in 5G terrestrial networks, aligning with our goal of integrating satellites with these networks. However, our solution is also applicable to the FDD case without any substantial modifications to the algorithm—the only adjustment needed is doubling the propagation delay to account for the additional CSI feedback step. Inspired by the Starlink satellites, we assume that the satellites are equipped with large UPAs. The downlink TPC is designed by considering the CSI uncertainty. We propose a DL-based solution for harnessing the delayed CSI for constructing a TPC matrix. Through this approach, we circumvent instantaneous channel prediction, hence reducing the complexity of the design while maintaining its adaptability to the time-varying environment. We employ deep reinforcement learning (DRL) for this purpose, and we harness the popular deep deterministic policy gradient (DDPG) technique for constructing our TPC matrix. The DRL agent experiences delays in observation and reward. Hence, a state augmentation technique is conceived for dealing with the delays. Our numerical results demonstrate the robustness of our proposed solution to channel delays for different handover mechanisms and different frequencies. Our contributions are contrasted with existing studies in the literature in Table I, while our detailed novelties are further elaborated on below.

- In this work we study the downlink transmission of a satellite equipped with a UPA to multiple single-antenna mobile users. We consider handovers in our design and propose a practical handover algorithm where the serving satellite is selected based on its distance to the centre of the coverage area. Furthermore, the frequency of handovers is used as a parameter in our numerical results.
- The proposed method circumvents conventional channel prediction, hence reducing the complexity, and constructs accurate TPC matrices despite relying on outdated CSI.
- In this work, DRL is employed for constructing the TPC matrix. DRL is particularly well-suited for erratically fluctuating environments like this one, where the system must account for the dynamically varying channels between a satellite and mobile users having random locations, while also handling handovers. This adaptability allows DRL to outperform traditional methods in the face of uncertainty in dynamic scenarios.
- The DRL agent employs the DDPG algorithm to handle the continuous action space, and uses a state augmentation technique for dealing with observation delays in the environment.
- Our numerical results include practical scenarios, relying on different frequencies, different handover mechanisms, and different pilot rates, in a dense four-layer Starlink constellation.

The rest of this paper is organized as follows. In Section II, the system model and the time-varying channel model are introduced. Section III presents the problem formulation, and Section IV is dedicated to our DRL-based solution. The constellation design, the details of the neural networks, and our simulation results are given in Section V, while Section VI concludes the paper. Throughout the paper, variables and functions are represented by italic lowercase letters, while constants are depicted by italic uppercase letters. Vectors and vector functions are denoted by bold lowercase letters, and matrices and matrix functions are represented by bold uppercase letters. The notation $|\cdot|$ indicates the absolute value of a variable, and $\mathrm{Tr}(\cdot)$ represents the trace function. Additionally, $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a vector/matrix, respectively. For clarity, the index for users is shown in square brackets in the superscripts of variables, e.g., $x^{[k]}$ indicating the variable x for the kth user.

	[6]	[13]	[14]	[15]	[16]	our work
Proposing precoding design	✓		✓	✓	✓	✓
Precoding design based on only stochastic CSI			✓	✓	✓	
Channel perdition based on delayed instantaneous CSI	✓	✓				
Precoding design based on current instantaneous CSI	✓					
Precoding design directly based on delayed CSI						✓
Consideration of handovers and optimizing handover scenario						✓
DRL-based precoding design, suitable for dynamic satellite-mobile communications						✓

Table I: Contrasting the novelty of our work with the relevant literature.

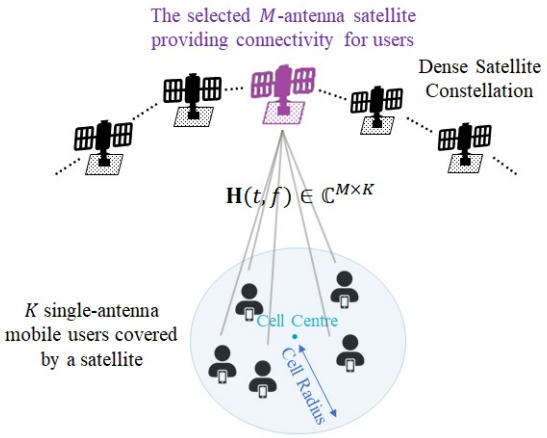

Figure 1: System Model: One satellite is providing coverage for K single-antenna users.

II. System Model

We aim to provide coverage for an off-the grid area on the ground that lacks terrestrial infrastructure, by directly connecting a LEO satellite to multiple single-antenna mobile handheld devices in this region. Eeach satellite in this LEO constellation is equipped with a UPA. We dedicate only a small subsection of a satellite's UPA to the downlink transmission of data for a specific region on earth containing K single-antenna users. The rest of the UPA resources can be dedicated to uplink data reception, downlink data transmission towards other areas on the ground, communication with other satellites or airborne vehicles, and so on. At each instant, multiple satellites are visible to the users, but only a single satellite is selected to provide coverage for that area. The system model is presented in Figure 1. As discussed in [5], selecting the nearest satellite to the users gives the best performance due to the lower path loss, albeit here the handover frequency becomes quite high. Thus, the trade-off between the handover frequency and the throughput should be investigated.

A. Satellite Handover Process

The satellite selection and the handover process are encapsulated by Algorithm 1. In this algorithm and throughout this paper, "time instant" refers to a specific moment in a discrete time span.

At a given time instant t, the satellite closest to the center of the coverage area is denoted as $S(t)$. The distance between this satellite and the center of the coverage area, measured at a specific time t, is represented by $d(S(t), t)$. If at a time instant t, we need to measure the distance between the satellite that was the nearest to the center of the coverage area at a previous time instant t', we denote it as $d(S(t'), t)$. Based on Algorithm 1, at a given time instant t, firstly, the satellite $S(t)$ is found. Then we evaluate if performing a handover would make a noticeable difference in the system quality. This is evaluated based on the distances of the satellites from the center of the coverage area, which results in their path loss. Given a small value of $\epsilon \in [0, 1)$, the handover only occurs if $\frac{d(S(t),t)}{d(S(t-1),t)} < 1 - \epsilon$. In other words, if this condition is met, at the time instant t, the satellite $S(t-1)$ is disconnected from the users, and the satellite $S(t)$ will provide connectivity for the users of the coverage area from this instant onward, until another handover occurs. When the parameter ϵ is set to 0, the system ensures that users within the coverage area are always connected to the nearest satellite, which typically provides the strongest signal. However, if ϵ is increased, the system becomes less sensitive to changes in signal strength. This means that handovers, or the switching from one satellite to another, happen less frequently. While this can lead to fewer interruptions caused by switching satellites, it also means that users may experience a higher average path loss.

Compared to other handover mechanisms, such as received power-based handover (RPH) and conditional handover (CH) techniques, our proposed method offers significant advantages in terms of simplicity, overhead reduction, and computational efficiency. The computational complexity of CH methods, such as the approach presented in [26], is significantly higher since it relies on solving an optimization problem and continuous monitoring of multiple parameters. In contrast, our technique is computationally lightweight, as it is based solely on predetermined geographical data rather than real-time optimization.

In the conventional RPH methods, a handover occurs when the received power from a target satellite (P_t) exceeds the received power from the serving satellite (P_s) by a predefined offset power value (P_{off}) for at least a specified duration (T_{off}). Mathematically, a handover is triggered when $P_t > P_s + P_{off}$ for at least T_{off} seconds. Through simulations, we observed that the channel gains and system sum rate achieved by our proposed method and the RPH technique are quite similar. Furthermore, the number of handovers and system performance in both methods can be tuned by adjusting ϵ in our case and P_{off}, T_{off} in the RPH approach. However, the key advantage of our proposed method is that it does not require additional user feedback, whereas RPH requires real-time feedback from users in an FDD scenario. Furthermore,

Algorithm 1 Satellite Selection and Handover at Time Step t

1. Select the nearest satellite to the center of the coverage area: $S(t)$
2. if $S(t) \neq S(t-1)$:
3. if $\frac{d(S(t),t)}{d(S(t-1),t)} < 1-\epsilon$:
4. Handover: Select $S(t)$ as the operating satellite
5. else:
6. Keep $S(t-1)$ as the operating satellite

our approach leverages predictable geographical locations, eliminating the need for continuous system monitoring, which is particularly advantageous for direct handheld-to-satellite connections.

B. Channel Model

Again, the selected satellite $S(t)$ is equipped with a UPA of $M = M_x \times M_y$ elements, which are spaced by half a wavelength from each other. Thus, the channel model presented in [13]–[15], [27] can be adopted. Throughout the remainder of the paper, the term "satellite" will be used for the selected satellite $S(t)$, while the index of the selected satellite $S(t)$ will be dropped from the equations. At a given time instant t, the kth user receives the following signal,

$$y^{[k]}(t,f) = \mathbf{h}^{[k]}(t,f)^H \mathbf{x}(t,f) + n^{[k]}(t,f), \tag{1}$$

where $\mathbf{h}^{[k]}(t,f) \in \mathbb{C}^{M\times 1}$ is the channel vector between the selected satellite and the kth user, f is the carrier frequency, $\mathbf{x}(t,f) \in \mathbb{C}^{M\times 1}$ is the signal vector transmitted by the satellite, and $n^{[k]}(t,f) \sim \mathcal{CN}(0,\sigma^2)$ is the additive white Gaussian noise at the kth user. The power of the receiver noise is $\sigma^2 = K_B \times T \times B$, where K_B denotes the Boltzmann constant, T is the equivalent noise temperature of the receiver in Kelvin, and B stands for bandwidth in Hz. The signal transmitted from the satellite is given by $\mathbf{x}(t,f) = \mathbf{V}(t,f)\mathbf{s}(t,f)$, where $\mathbf{V}(t,f) = [\mathbf{v}^{[1]}(t,f), ..., \mathbf{v}^{[k]}(t,f)] \in \mathbb{C}^{M\times K}$ is the TPC matrix and $\mathbf{s}(t,f) = [s^{[1]}(t,f), ..., s^{[K]}(t,f)]^T \in \mathbb{C}^{K\times 1}$ is the vector of symbols. The channel vector $\mathbf{h}^{[k]}$ is formulated as [28],

$$\mathbf{h}^{[k]}(t,f) = \frac{\mathbf{h}^{[k]}_{\text{LOS}}(t,f) + \mathbf{h}^{[k]}_{\text{NLOS}}(t,f)}{\text{FSPL}^{[k]}(t,f)}, \tag{2}$$

where $\text{FSPL}^{[k]}$ is the free-space path-loss between the satellite and the kth user, which is calculated by $\text{FSPL}^{[k]} = \frac{4\pi d^{[k]}(t) f}{c}$. Here, $d^{[k]}(t)$ represents the distance between the satellite and the kth user at the time instant t, and c stands for the speed of light. Furthermore, $\mathbf{h}^{[k]}_{\text{LOS}}$ is the line of sight (LOS) component and $\mathbf{h}^{[k]}_{\text{NLOS}}$ represents the stochastic non-line-of-sight (NLOS) component of the CSI. These components are given by (3) and (4) at the top of the next page.

The notations in (3) and (4) for the time instant t and frequency f are as follows: $P^{[k]}(t)$ is the number of NLOS paths of user k, and $\kappa^{[k]}(t)$ is the Rician factor for the link between the kth user and the satellite. The notations $\bar{v}^{[k]}_{\text{Sat}}(t,f)$ and $\bar{v}^{[k]}_{\text{UE}}(t,f)$ represent the Doppler frequency shifts caused by the motion of either the satellite (superscript Sat) or the user (superscript UE) for the LOS link between the satellite and the kth user. Also, $v^{[k]}_{\text{Sat},p}(t,f)$ and $v^{[k]}_{\text{UE},p}(t,f)$ stand for the Doppler frequency shifts in the pth NLOS path caused by the motion of the satellite and the kth user, respectively. The delay of the LOS path is given by $\tau^{[k]}_{\text{LOS},l}(t)$, while that of the pth NLOS path is denoted by $\tau^{[k]}_{\text{NLOS},p}(t)$. Also, $\theta^{[k]}_{\text{LOS}}(t)$ and $\theta^{[k]}_{\text{NLOS},p}(t)$ stand for the angles of horizontal directions for the LOS and the pth NLOS paths, respectively [14]. The angles for the vertical directions of the LOS and the pth NLOS paths are given by $\psi^{[k]}_{\text{LOS}}(t)$ and $\psi^{[k]}_{\text{NLOS},p}(t)$. The array response vector under the UPA model is $\mathbf{u}(\theta,\psi) = \mathbf{a}(\cos(\theta)\sin(\psi), M_x) \otimes \mathbf{a}(\cos(\psi), M_y)$, where

$$\mathbf{a}(\phi, N) = \frac{1}{\sqrt{N}}[1, e^{-j2\pi\frac{d_s}{\lambda}\phi}, ..., e^{-j2\pi\frac{d_s}{\lambda}(N-1)\phi}]^T \in \mathbb{C}^{N\times 1}, \tag{5}$$

is the one-dimensional steering vector function for a uniform linear array (ULA). In (5), $\lambda = \frac{c}{f}$ stands for the carrier wavelength and d_s is the antenna spacing [27].

The Doppler frequency shifts resulting from the satellite movement remain constant across the LOS and the NLOS paths due to the high altitude of the satellite. This means that at a given time instant t, we have $v^{[k]}_{\text{Sat},p}(t,f) = \bar{v}^{[k]}_{\text{Sat}}(t,f)$, $\forall p \in \{1, ..., P^{[k]}(t)\}$. The Doppler frequency shift of the satellite can be determined by $\bar{v}^{[k]}_{\text{Sat}}(t,f) = \frac{q}{c} f \cos(\omega^{[k]}(t))$, where q represents the satellite's velocity and $\omega^{[k]}(t)$ stands for the angle between the satellite's movement trajectory and its LOS path to user k. However, the Doppler frequency shifts due to user movements vary for each path [15]. Given that the distance between satellites and users substantially exceeds the scale of scattering, the angular spread of NLOS paths can be deemed negligible. Consequently, for a designated satellite and user k, it is reasonable to assume that $\theta^{[k]}_{\text{NLOS},p}(t) = \theta^{[k]}_{\text{LOS}}(t) = \theta^{[k]}(t)$ and $\psi^{[k]}_{\text{NLOS},p}(t) = \psi^{[k]}_{\text{LOS}}(t) = \psi^{[k]}(t)$, $\forall p \in \{1, ..., P^{[k]}(t)\}$ [13].

III. Problem Formulation

By relying on the TDD technique, the CSI is estimated in the uplink, and the channel estimates are used for precoding in the downlink. However, due to the propagation delay and the limited coherence interval, especially at higher frequencies, the outdated CSI problem occurs. In this context, we aim to maximize the achievable rate by optimizing the TPC matrix under the outdated CSI scenario. To do this, first we calculate the achievable rate, and then we formulate the optimization problem. The achievable rate is calculated in Appendix A. We now formulate the rate maximization problem as

$$\begin{aligned} \max_{\mathbf{V}(t,f)} \quad & \sum_{k=1}^{K} \log\left(1 + \frac{|\mathbf{h}^{[k]H}(t,f)\mathbf{v}^{[k]}(t,f)|^2}{\sum_{i\neq k}^{K} |\mathbf{h}^{[k]H}(t,f)\mathbf{v}^{[i]}(t,f)|^2 + \sigma^2}\right) \\ \text{s.t.} \quad & \text{Tr}\left(\mathbf{V}(t,f)\mathbf{V}^H(t,f)\right) \leq P, \end{aligned} \tag{6}$$

where P represents the power budget of the satellite.

Aside from the non-convex nature of this problem, which makes it challenging to solve, the primary issue with the opti-

$$\mathbf{h}_{\text{LOS}}^{[k]}(t,f) = \sqrt{\frac{\kappa^{[k]}(t)}{1+\kappa^{[k]}(t)}} \times exp\left(j2\pi t\left(\bar{v}_{\text{Sat}}^{[k]}(t,f) + \bar{v}_{\text{UE}}^{[k]}(t,f)\right)\right) \times exp\left(-j2\pi f \tau_{\text{LOS}}^{[k]}(t)\right) \times \mathbf{u}\left(\theta_{\text{LOS}}^{[k]}(t), \psi_{\text{LOS}}^{[k]}(t)\right), \quad (3)$$

$$\mathbf{h}_{\text{NLOS}}^{[k]}(t,f) = \sqrt{\frac{1}{P^{[k]}(t)\left(1+\kappa^{[k]}(t)\right)}} \times \sum_{p=1}^{P^{[k]}(t)} g_p^{[k]}(t) \times exp\left(j2\pi t\left(v_{\text{Sat},p}^{[k]}(t,f) + v_{\text{UE},p}^{[k]}(t,f)\right)\right) \times exp\left(-j2\pi f \tau_{\text{NLOS},p}^{[k]}(t)\right) \times \mathbf{u}\left(\theta_{\text{NLOS},p}^{[k]}(t), \psi_{\text{NLOS},p}^{[k]}(t)\right), \quad (4)$$

mization in (6) lies in the delayed CSI. A promising approach to tackle this is by establishing a distribution for the CSI uncertainty and conceiving a robust TPC technique, similar to the strategies outlined in [29], [30]. This strategy was also applied in satellite communication [31], where the estimated channel was assumed to be the sum of the actual channel value and a certain error. Nevertheless, these approaches have been devised for terrestrial communication systems, where uncertainty often stems from user movements. Consequently, the variance of CSI uncertainty is relatively low compared to the real channel strength. As illustrated in [29], [30] and other related literature, as the uncertainty increases, these methods become less effective. In essence, the conventional robust designs employed in terrestrial communications are capable of compensating for the performance degradation resulting from channel estimation errors up to a certain threshold.

In satellite communications, the primary source of channel uncertainty is the delayed CSI, which depends on factors such as the user-to-satellite distance, frequency, bandwidth, and transmission rate. Consequently, the channel estimation error cannot be adequately represented by a Gaussian distribution—But even if it could be, the variance of this error would be excessive for traditional terrestrial communication methods. As demonstrated in [31], the delayed CSI issue can be addressed using conventional schemes for frequencies up to 1 GHz. Therefore, our objective is to leverage the correlations between channels at different time instances to address the challenges posed by delayed CSI.

IV. DRL-Based Precoding Solution

We employ the DRL algorithm to solve the optimization problem presented in (6). The solution to this problem heavily relies on the time-variant CSI acquired. The dynamic and continuous nature of the CSI observations during communication between a satellite and a mobile user makes DRL more suitable for this problem compared to supervised learning. It is important to note that in this communication scenario, the channel states are influenced by factors such as the users' locations, their speeds, antenna patterns, and other user-specific information that is not accessible to the satellites and can vary unpredictably. Therefore, DRL is employed which can exploit the correlation of the CSI throughout time and construct accurate TPC matrices. Furthermore, DRL can readily handle the delayed CSI. In [32], reinforcement learning (RL) has been applied to scenarios involving agents associated with deterministic and stochastic observations and/or action delays. Our case study focuses on an environment with only deterministic observation delays. In this context, satellites can estimate the delay based on their approximate distance to the users, allowing them to determine the number of time intervals required for observations to reach them. Consequently, the CSI has a deterministic delay model.

In a Reinforcement Learning (RL) setup, an agent interacts with its environment over discrete time steps. This interaction is described as a Markov Decision Process (MDP). At each time step t, the agent chooses an action $\mathcal{A}_t$ based on the current state $\mathcal{S}_t$ it observes and a predefined policy denoted as π. Upon taking this action, the agent receives a reward $\mathcal{R}_t$, and transitions from the current state $\mathcal{S}_t$ to the next state $\mathcal{S}_{t+1}$. It is worth noting that the optimization variable for the problem in (6) is the TPC matrix, which exists in a continuous space. Consequently, if we use DRL to tackle this problem, our action space must also be continuous, since the action involves selecting the optimal TPC strategy for maximizing the achievable rates. To effectively harness DRL in scenarios of continuous action spaces, we employ the DDPG technique, which is a model-free, off-policy actor-critic algorithm capable of operating in such environments. DDPG integrates the strengths of both deep Q-networks (DQN) [33] and policy gradient (PG) techniques [34], making it eminently suitable for managing continuous and high-dimensional action spaces [35].

A. Basics of DDPG

In DDPG, the actor network functions as a deterministic policy network that selects actions from a continuous action space $\mathcal{A}$. This policy is denoted as $\mu : \mathcal{S} \rightarrow \mathcal{A}$, where $\mu(\mathcal{S};\boldsymbol{\theta}_\mu)$ represents the action chosen by the actor network given the state $\mathcal{S}$, and $\boldsymbol{\theta}_\mu$ denotes the network parameters. The critic network acts as a Q network $Q(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_q)$, with $\boldsymbol{\theta}_q$ representing its parameters. The critic assesses the actor's action by assigning it a Q-value, and the primary objective of DDPG is to maximize this Q-value. Similar to DQN, DDPG uses experience replay for mitigating the correlations among different training samples. Additionally, to compute the corresponding target values, duplicate actor and critic networks, also referred to as target networks, are created. These are denoted by $\mu^*(\mathcal{S};\boldsymbol{\theta}_{\mu^*})$ and $Q^*(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_{q^*})$, respectively.

We realize $\mu(\mathcal{S};\boldsymbol{\theta}_\mu)$ and $Q(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_q)$ as evaluation networks, which have the same structure as the target networks, but with different parameters. By using the following soft

update, the parameters of the target networks are calculated as

$$\boldsymbol{\theta}_{j^*} = \tau\boldsymbol{\theta}_j + (1-\tau)\boldsymbol{\theta}_{j^*},\ j \in \{\mu, q\},\ \tau << 1. \quad (7)$$

In DDPG, the exploration is treated independently from the learning process. More explicitly, an exploration policy of $\mu^{'}(\mathcal{S}) = \mu(\mathcal{S};\boldsymbol{\theta}_\mu) + \mathcal{N}$ may be formulated, where the noise process $\mathcal{N}$ is appropriately chosen to match the environment.

B. DRL formulation with deterministic observation delays

To develop our DRL model, we first define our MDP elements, including the observation, action as well as reward, and then explain our approach in detail.

1) Action Space

The continuous action space is defined in terms of the imaginary and the real components of the TPC matrix $\mathbf{V}$, reshaped to form a $1 \times 2MK$ vector, where the first MK elements represent the real components of $\mathbf{V}$ and the second part is constituted by the imaginary components. The upper and lower bounds of the action space are defined by the power constraint in (6). In other words, once the action is selected by the actor network and a noise component is added to construct the exploration ($\mu^{'}(\mathcal{S}) = \mu(\mathcal{S};\boldsymbol{\theta}_\mu) + \mathcal{N}$), the legal action is determined by projecting the output $\mu^{'}(\mathcal{S})$ onto a ball centered at the origin and having the radius of P [1].

2) Observation Space

In a typical Markov Control Model having Perfect State Information (MCM-PSI), the decision-maker is assumed to have access to the current state of the system at each decision instant. This ensures that the state transitions are only dependent on the current state and the chosen action, thus satisfying the Markov property. However, in scenarios where there is a delay in state observation, this perfect information assumption no longer holds. The state available to the decision-maker is outdated, leading to what is known as a Markov Control Model with Imperfect State Information (MCM-ISI) [36]. To address this challenge and restore the Markov property, the state space can be augmented. Specifically, the augmented state should include not only the most recent known state but also all actions taken during the delay period. This approach effectively transforms the delayed information problem into one where the decision-maker has sufficient information to make decisions as if the state were perfectly known. In [36] it is demonstrated that by enlarging the state space to include both the most recent known state and the sequence of actions taken during the delay, a Markov control model with N-Step Delayed State Information (N-SDSI) can be effectively transformed into an MCM-PSI. This transformation ensures that the decision-maker has all relevant information to make optimal decisions, thus restoring the Markov property.

[1]The projection of a point $\mathbf{X}$ into a set $\mathbb{S}$ is defined by $\min_{\mathbf{P}\in\mathbb{S}} \|\mathbf{X}-\mathbf{P}\|$. In case $\mathbb{S}$ is a sphere centered at the origin with radius P ($\mathbb{S} = \{\mathbf{X} | \|\mathbf{X}\| \le P\}$), the projection of $\mathbf{X}$ is obtained by $P\frac{\mathbf{X}}{\|\mathbf{X}\|+\max(0,P-\|\mathbf{X}\|)}$.

At the time instant t, the agent receives the CSI from T_d time steps ago; hence, we have

$$\mathbf{H}(t-T_d, f) = \left[\mathbf{h}^{[1]}(t-T_d, f), ..., \mathbf{h}^{[K]}(t-T_d, f)\right]. \quad (8)$$

The duration of time steps is set to Δt. The value of T_d is determined by $T_d = \frac{c\Delta t}{d}$, where d is the distance between the satellite and the center of the coverage area and c is the speed of light. Based on [32], [36], a given Constant Delay MDP (CDMDP) with T_d observation delays, denoted by $< \mathcal{S}, \mathcal{A}, r, T_d >$, with r as the reward, may become equivalent to an MDP $< \mathcal{S}^{'}, \mathcal{A}, r, 0 >$ if the observation space is augmented as

$$\{s(t-T_d), a(t-T_d), a(t-T_d+1), ..., a(t-1)\} \in \mathcal{S}^{'}, \quad (9)$$

where $s(t-T_d)$ is the delayed observation from the environment, and $a(t-T_d), a(t-T_d+1), ..., a(t-1)$ are all the actions taken during the delay period. In our case-study, the observation space at time instant t includes

$$s_t = \{\mathbf{H}(t-T_d, f), \mathbf{V}(t-T_d), ..., \mathbf{V}(t)\}. \quad (10)$$

In other words, the size of our observation space is $1 \times 2(T_d + 2)MK$.

3) Reward

Consider our case study, where the problem is an MDP with an observation delay of T_d. At a given time instant t, the agent selects an action $a(t)$, but it cannot receive the reward until the time step $t + T_d$. Thus, a delay in observation results in a delay in receiving the reward as well. In our DRL model, the reward is calculated based on the achievable rate. Given that the CSI is delayed by T_d time steps, the reward at any time t is a function of the delayed CSI, $\mathbf{H}(t-t_d, f)$, and the TPC matrix derived from the action taken T_d time steps earlier, $\mathbf{V}(t-T_d)$. As demonstrated in [32], maximizing a modified reward structure that depends on the state and action at $t-T_d$, i.e., $s(t-T_d)$ and $a(t-T_d)$, is equivalent to maximizing a conventional reward structure that is solely a function of the current state and action, $s(t)$ and $a(t)$.

At first, a continuous value is calculated for the reward by (11) at the top of the next page. Then, for the purpose of better convergence, this reward is quantized and calculated by (12) on top of the next page. In (12) $\lceil . \rceil$ represents the ceiling function, and η_1, η_2 are threshold values. Note that in order to improve the speed of convergence, we present the agent with more reward if there is any rate improvement compared to the previous time step.

It is worth mentioning that due to the structure of this reward, which focuses on maximizing the sum rate, the algorithm is inherently opportunistic, and user fairness is not a primary objective. However, by modifying the reward function and setting a lower bound for the minimum rate among users, we can ensure a guaranteed quality of service for each user. As an example, the following fairness-aware reward structure can be adopted:

$$r_{\text{fair}}(t) = r(t) + \zeta, \quad (13)$$

$$r_{con}(t) = \sum_{k=1}^{K} \log\left(1 + \frac{|\mathbf{h}^{[k]H}(t-T_d,f)\mathbf{v}^{[k]}(t-T_d,f)|^2}{\sum_{i\neq k}^{K}|\mathbf{h}^{[k]H}(t-T_d,f)\mathbf{v}^{[i]}(t-T_d,f)|^2+\sigma^2}\right). \tag{11}$$

$$r(t) = \begin{cases} \max(\lceil r_{con}(t)-\eta_1\rceil,0)-\eta_2+1, & r_{con}(t) > r_{con}(t-1) \\ \max(\lceil r_{con}(t)-\eta_1\rceil,0)-\eta_2, & otherwise \end{cases} \tag{12}$$

where

$$\zeta = \begin{cases} \zeta_1, & \min\{r_{con}^{[1]}(t)\} < \nu_1, \\ \zeta_2, & \min\{r_{con}^{[1]}(t)\} < \nu_2, \\ \zeta_3, & \min\{r_{con}^{[1]}(t)\} < \nu_3, \\ \zeta_4, & \min\{r_{con}^{[1]}(t)\} < \nu_4. \end{cases} \tag{14}$$

The values of ζ_i and ν_i, where $i \in \{1, ..., \text{Number of Cases} = 4\}$, are chosen based on system requirements and the desired minimum rate guarantee for each user, such that $0 > \zeta_1 > \zeta_2 > \zeta_3 > \zeta_4$ and $\nu_1 > \nu_2 > \nu_3 > \nu_4 > 0$. This modification introduces a soft fairness constraint, ensuring a baseline quality of service while maintaining sum-rate maximization.

C. The overall algorithm

The DDPG-based DRL algorithm used for solving (6) is detailed in Algorithm 2. First, the parameters of the DRL are selected, and both the target as well as the evaluation actor and critic networks are generated with the aid of randomly assigned weights. Then, an experience replay buffer with capacity C is built to store the transitions. Our dataset includes the channel gains, which are modeled based on the locations of satellites in a constellation, and it includes T_{total} time steps. These pieces of information are used then in Z episodes, each including J time steps. In the first T_d time steps, where the satellite has not received any pilots in its uplink and therefore cannot estimate the CSI, the actions are selected randomly. Hence, the learning starts at $t = T_d + 1$. At the beginning of each time step, the delayed channel is estimated, and based on (10), the observation space is formed. The action is selected via the actor network, and for the purpose of exploration, a continuous action noise, having the power of $\sigma_{\mathcal{N}}^2$ is added to it. Note that the power of noise in the first time steps is high, but it is gradually reduced for enforcing exploitation over exploration. The constraint in problem (6) is satisfied via projection, and then the legal action is transformed into a complex TPC matrix $\mathbf{V}$. The reward is then calculated using (12) and the entire transition is stored in the buffer. A mini-batch with size T_B is then selected from the buffer for calculating the Q-value q_j as

$$q_j = \begin{cases} r_j, & j = T_B, \\ r_j + \alpha Q^*(s_{j+1}, \mu^*(s_{j+1};\boldsymbol{\theta}_{\mu^*});\boldsymbol{\theta}_{q^*}), & j < T_B, \end{cases} \tag{15}$$

where α represents the critic learning rate. The loss function for the critic evaluation network is given by

$$L_{\text{critic}}(\boldsymbol{\theta}_q) = \frac{1}{T_B}\sum_{j=1}^{T_B}(q_j - Q(s_j,a_j;\boldsymbol{\theta}_q))^2. \tag{16}$$

The critic evaluation network is updated by minimizing the critic loss function. The actor network is updated by minimizing the following loss function

$$L_{\text{actor}}(\boldsymbol{\theta}_\mu) = -\frac{1}{T_B}\sum_{j=1}^{T_B} Q\left(s_j, \mu(s_j;\boldsymbol{\theta}_\mu);\boldsymbol{\theta}_q\right). \tag{17}$$

The actor evaluation network is updated using the PG technique relying on the ascend factor

$$\Delta_{\boldsymbol{\theta}_\mu} = \frac{1}{T_B}\sum_{j=1}^{T_B}\Big(\nabla_a Q(s_j,\mu(s_j;\boldsymbol{\theta}_\mu);\boldsymbol{\theta}_q)|_{s=s_j,a=\mu(s_j)} \nabla_{\boldsymbol{\theta}_\mu}\mu(s_j;\boldsymbol{\theta}_\mu)|_{s_j}\Big). \tag{18}$$

Finally, the target networks are updated by soft update in (7).

Algorithm 2 The DRL-based solution using DDPG algorithm

Input : discount factor λ, soft update coefficient τ, buffer capacity C, batch size T_B, actor learning rate β, critic learning rate α

Initialization :

Randomly initialize the parameters of the four networks $\mu(\mathcal{S};\boldsymbol{\theta}_\mu)$, $Q(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_q)$, $\mu^*(\mathcal{S};\boldsymbol{\theta}_{\mu^*})$, $Q^*(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_{q^*})$.
Empty the experience replay buffer.
Initialize $\mathcal{N}$ (with variance $\sigma_{\mathcal{N}}^2$) for action exploration.
Set time to $t = T_d + 1$.
Initialize random actions for the first T_d time steps.
for episode = 1 : Z **do**
  Reset the environment, receive the initial observations
  for n = 1 : J **do**
    1. Obtain outdated channel $\mathbf{H}(t-T_d,f)$ and form the observation space using (10)
    2. Select the action $a_t = \mu(s_t;\boldsymbol{\theta}_\mu) + \mathcal{N}$.
    3. Obtain the legal action using projection.
    4. Store a_t into the matrix $\mathbf{V}$.
    5. Calculate the reward via (12) and store the transition $\{s_t, a_t, r_t, s_{t+1}\}$.
    6. Sample a mini-batch with size T_B from the replay buffer as $\{s_j, a_j, r_j, s_{j+1}\}$.
    7. Determine the target Q-value via (15).
    8. Update $Q(\mathcal{S},\mathcal{A};\boldsymbol{\theta}_q)$ by minimizing (16).
    9. Update $\mu(\mathcal{S};\boldsymbol{\theta}_\mu)$ using the sampled policy gradient in (18).
    10. Update the target networks using (7).
    11. t = t+1
  end for
end for

Note that according to this algorithm, the size of the neural networks inherently depends on the number of users (K).

Naturally, this limits the number of users that can be supported by the system. To address this scalability issue, in our study, we allocated a small number of antennas ($M = 9$) to serve a specific terestrial region. This approach allows the system to focus on a smaller subset of users within that region, while other users are served by different antennas of the same satellite. This design leverages spatial multiplexing to maximize the overall system capacity without significantly increasing the complexity of the neural networks. Furthermore, to serve a larger number of users within the same terestrial region, multiple satellites can be utilized to form a large-scale distributed satellite network. By employing distributed machine learning techniques such as multi-agent reinforcement learning, the scalability challenge can be effectively addressed. This remains an area for future research.

V. Numerical Results

In this section, we evaluate the performance of the proposed system through simulations. We assume that enough satellites are in orbit, so a satellite is visible to users on the ground, at each time instant. The application of the method presented here is not limited to a specific satellite type or formation; in fact, the constellation may be part of a space-air-ground integrated network (SAGIN).

A. System parameters

For our simulations, we consider a constellation of satellites, each containing a UPA of $M = 3 \times 3$ elements dedicated to supporting $K = 2$ single-antenna users. In our simulations, the users are considered to be mobile handheld devices moving at random speeds of up to 3 m/s in random directions. The users are located in the Lake District National Park, UK, which has limited terrestrial infrastructure, especially in its off-the-grid locations. The area contains users in a 40 km radius of a given location with the latitude of 54.5260000° and the longitude of -3.3000000°. We use a constellation similar to Starlink, and the parameters of this design are given in Table II. The transmission frequency for our study is $f = 2$ GHz and the bandwidth is $BW = 0.02f$. The minimum channel coherence interval is calculated by $T_c^{\min} = \frac{c\sqrt{R_E + d_{\text{alt}}^{\min}}}{f\sqrt{GM_E}\cos(\vartheta^{\min})}$, where $R_E = 6.371 \times 10^6$ (m) is the radius of the earth, $d_{\text{alt}}^{\min} = 540$ km is the minimum altitude of a satellite connected to the users, $G = 6.674 \times 10^{-11}$ is the gravitational constant, $M_E = 5.972 \times 10^{24}$ is the mass of the earth, and $\vartheta^{\min} \approx 80^{\circ}$ is the minimum elevation angle of a satellite transmitting to the users. Thus, the minimum coherence interval is about $T_c^{\min} = 115$μsec. Note that a coherence interval is a time interval in which the channel may be considered near-constant. Hence, it is plausible that there is a correlation between the channels in consecutive time intervals. The propagation delay, τ_d, is calculated by the distance between the satellite and the user divided by the speed of light, and it is roughly equal to $\tau_d \approx 1.9$ msec. However, the number of time steps for the observations to arrive at the satellite is calculated by $T_d = \frac{\tau_d}{\Delta t}$, where Δt is the time difference between two consecutive pilot transmissions. In our simulations, we set $\Delta t = T_c$ and $\Delta t = \tau_d$ to compare the results.

Parameters	Layer 1	Layer 2	Layer 3	Layer 4
Num. orbital planes	72	36	6	72
Num. satellites per plane	22	20	58	22
Altitude	550 km	570 km	560 km	540 km
Inclination angle	53°	70°	97.6°	53.2°

Table II: Parameters of the constellation with four layers of satellites in orbit.

Actor Layers	Number of Neurons	Activation
Input layer	Num. States	
Hidden layer 1	Num. Actions	relu
Hidden layer 2	Num. Actions	relu
Hidden layer 3	Num. Actions	relu
Hidden layer 4	Num. Actions	relu
Output layer	Num. Actions	tanh

Table III: Actor network

The horizontal and vertical direction angles for the waveform w.r.t. the ULA are realized by uniform random variables in the range of $\theta^{[k]}(t), \psi^{[k]}(t) \in [\vartheta^{[k]}(t), \pi - \vartheta^{[k]}(t)]$, where $\vartheta^{[k]}(t)$ is the elevation angle of the satellite relative to the kth user. The number of NLOS paths between the kth user and the satellite $P^{[k]}(t)$, is selected randomly by using a discrete uniform distribution on the intervals of $[2, 7]$. Similarly, the Rician factor $\kappa^{[k]}(t)$ is randomly chosen from a discrete uniform distribution within the range $[81, 90]$. It is assumed that the satellite's power budget for this transmission is $P = 1$ W, while the receiver noise power σ^2 is calculated at a temperature of $T = 280$ K.

B. Neural network parameters

The parameters of the actor and critic neural networks are given in Table III and Table IV, respectively. Note that the number of actions is Num. Actions $= 2MK = 36$, but the size of the observation space depends on T_d and it is equal to Num. States $= 2(T_d + 2)MK$. The actor network consists of four hidden layers, all of which have the same size as the output layer. By contrast, the critic network has hidden layers of varying sizes. Since the critic network's output layer consists of a single neuron, but its input layer has a large number of neurons, the first two hidden layers are designed with a higher number of neurons. The number of neurons is then gradually reduced in each subsequent layer, ensuring a smooth transition toward the single-neuron output. This structure was determined through extensive simulations, where we found that this gradual reduction in neuron count optimally balances the learning performance vs. the computational efficiency for our system. In this paper, an episode is a set including $J = 480$ time steps. For the discount factor λ, selecting a value close to 1 promotes long-term reward optimization, while lower values focus on immediate rewards. We selected $\lambda = 0.95$ as it provides a balance between stability and convergence speed. The soft update coefficient τ determines how quickly the target networks are updated. By selecting a small value of $\tau = 0.005$, we reduce drastic changes to the target networks and enhance

Critic Layers	Number of Neurons	Activation
State input layer	Num. States	
Action input layer	Num. Actions	
Concatenation	Num. States+Num. Actions	
Hidden layer 1	2 * Num. Actions	relu
Hidden layer 2	3.46 * Num. Actions	relu
Hidden layer 3	1.8 * Num. Actions	relu
Hidden layer 4	0.96 * Num. Actions	relu
Hidden layer 5	0.54 * Num. Actions	relu
Hidden layer 6	0.26 * Num. Actions	relu
Output layer	1	None

Table IV: Critic network

the stability of the training. The buffer capacity is set to $C = 50000$ and the batch size of $T_B = 64$ is selected to offer stable learning without excessive computational complexity. As for the actor and critic learning rates, their values are usually selected such that the critic learning rate (α) is greater than the actor learning rate (β) since the critic network needs to adapt more rapidly. We found that the values of $\alpha = 0.002$ and $\beta = 0.001$ provide a stable training. The noise process added to the action for the purpose of exploration is an AWGN noise process with the initial power of $\sigma^2_{\mathcal{N}} = 0.11$. However, this noise power is tapered off in each time step by a factor of 0.99996 until it reaches $\sigma^2_{\mathcal{N}} = 0.05$. Finally, the reward thresholds in (12) are given by $\eta_1 = 4$, $\eta_2 = 2$.

We used Matlab for generating the satellite movements, selecting the satellite in service via Algorithm 1, and generating the CSI throughout time. The output CSI was then used as a .mat matrix of size ($M = 9 \times$ Num. time steps = 2000000 $\times$ $K = 2$) in Python, where the DRL agent was programmed using TensorFlow.

C. Algorithm performance evaluation

Figure 2 depicts the episodic learning process of the agent in the case where the pilots are received every $\Delta t = T_c = 115$μsec. Through $Z = 416$ episodes, each including $J = 480$ time steps, we evaluate the system performance within a time span of 23 seconds. This figure presents three case studies examining the impact of pilot signal delays on the system performance:

1) Perfect CSI Case ($T_d = 0$):
   - In this scenario, we assume that pilot signals are received without any delay.
   - This represents the upper bound of system performance, as the CSI is always up-to-date.
2) Minimal Delay Case ($T_d = 1$):
   - Here, pilot signals experience a slight delay of one time step before reaching the receiver.
   - This case allows us to analyze how a small delay in CSI updates affects the system performance.
3) Realistic Delay Case ($T_d = 16$):
   - This scenario models a practical situation where the pilots experience a significant delay of 16 time steps before arriving.
   - The delay is calculated based on the estimated propagation delay of $\tau_d = 1.9$ msec, using the time step duration of 115 μsec, leading to $T_d = \frac{1.9\text{msec}}{115\mu\text{sec}} \approx 16$ time steps.

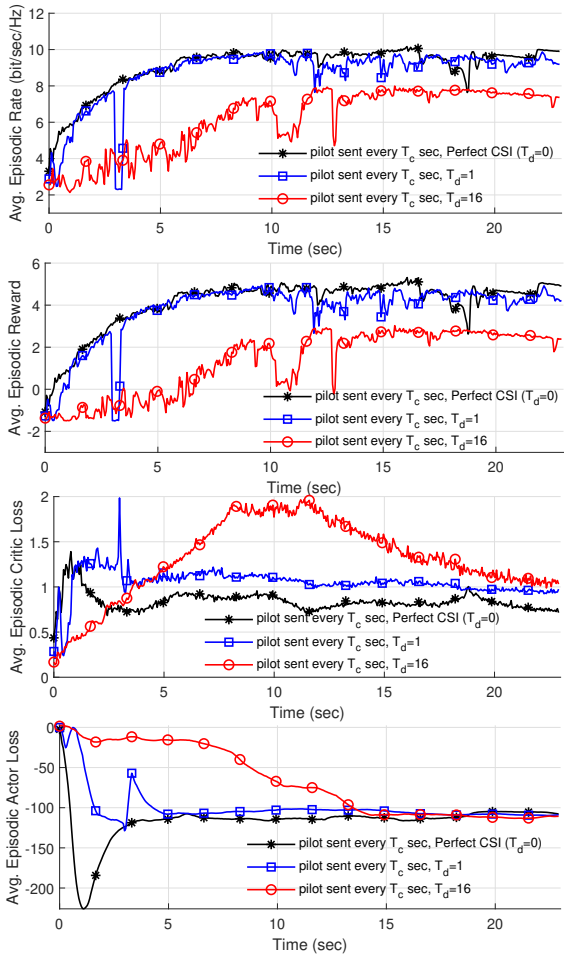

Figure 2: Learning process for an agent receiving pilots every T_c sec in three scenarios of perfect CSI, $T_d = 1$ and $T_d = 16$ time steps.

This comparison helps to evaluate the robustness of the system under different CSI delay conditions, providing insight into the trade-offs between the system's responsiveness and real-world constraints. As shown in the figure, the system performs better in terms of both reward and achievable rate, when the CSI is perfect and when the time delays are minimal. Additionally, an analysis of the actor and critic losses reveals that the system converges more promptly for shorter delays, particularly when $T_d = 0$ or $T_d = 1$. The slower convergence observed in the more realistic scenario with $T_d = 16$ is attributed to delayed reward feedback, which forces the agent to learn at a slower pace. In the case of perfect CSI, the agent benefits from immediate and accurate CSI feedback, allowing for faster convergence. The sharp drop in actor loss at the beginning is an immature convergence, which is followed by further exploration before the algorithm stabilizes. This behavior is influenced by the reward function, which favors sum-rate improvements over consecutive time steps ($r_{con}(t) > r_{con}(t-1)$), causing higher Q-values in the initial stages and consequently lower actor loss values.

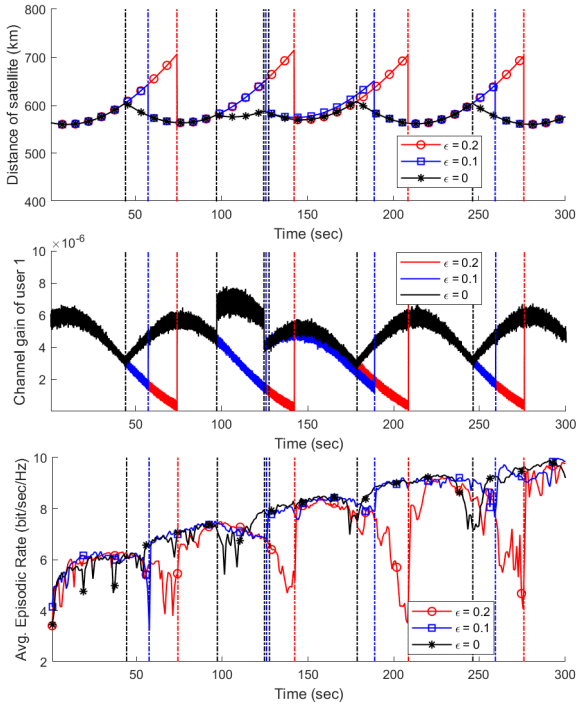

Figure 3: Handover analysis in three cases of $\epsilon \in \{0, 0.1, 0.2\}$. The pilots are sent every τ_d sec, and the achievable rates are presented under the assumption of perfect CSI.

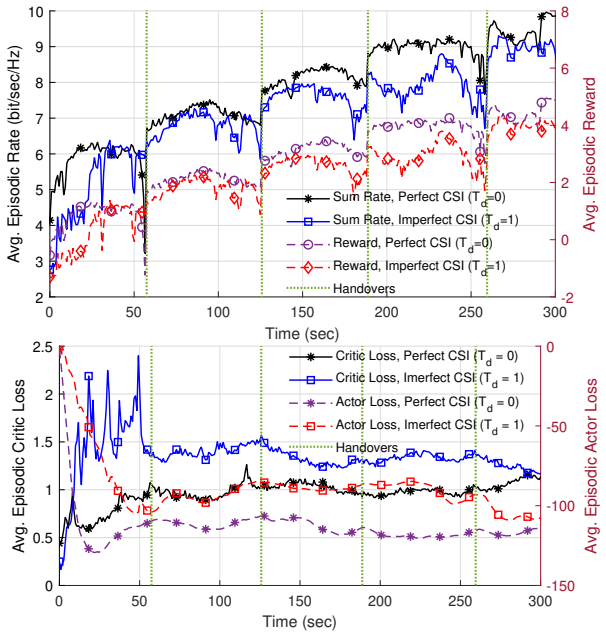

Figure 4: Learning process for an agent with $\epsilon = 0.1$ receiving pilots every $\tau_d = 1.9$ msec, in two scenarios of perfect CSI and imperfect CSI with $T_d = 1$.

Note that during the 23 sec time span of Figure 2, no handover occurs. In a high pilot rate scenario like this, the agent gets to interact with the environment in more time steps and learns more before a handover happens. Furthermore, on one hand, the correlation of channels through time is more

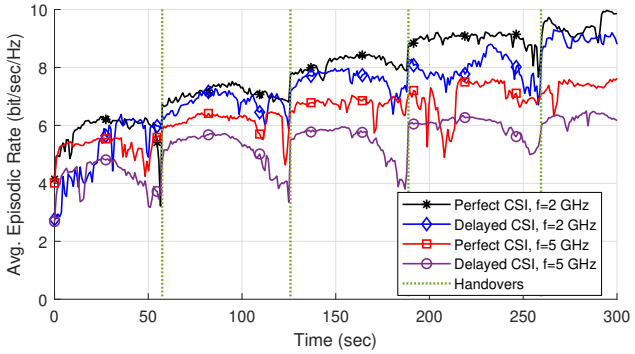

Figure 5: Achievable rate for an agent with $\epsilon = 0.1$, receiving pilots every τ_d seconds, in two scenarios of perfect and imperfect CSI, for two operating frequencies of $f = 2$ GHz and $f = 5$ GHz, with $BW = 0.02f$ in both.

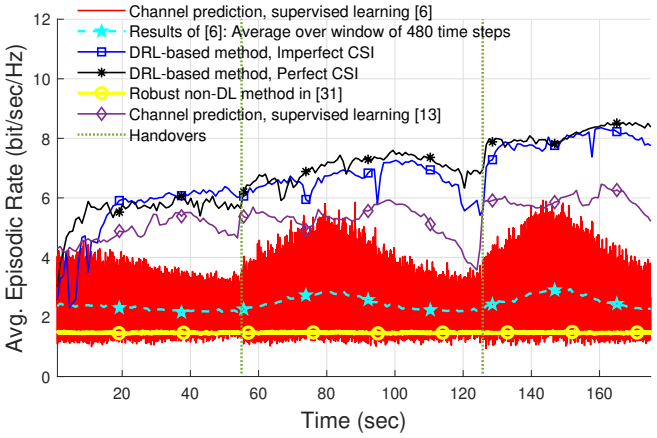

Figure 6: Comparison between our proposed scheme and other methods in the literature, in a system with $f = 2$ GHz and $BW = 0.02f$.

pronounced; hence, the agent can learn these correlations better. On the other hand, the value of T_d in a high pilot rate scenario is high, which increases the size of the observation space. Thus, the learning process will be more complex for the agent, which can result in slower convergence and lower rewards.

To this end, the effect of pilot rate on the system is investi-

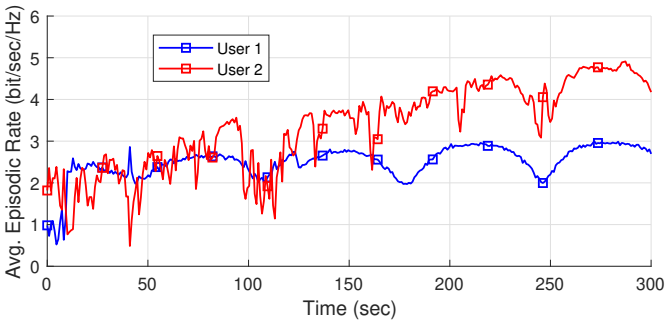

Figure 7: Average rate per user in an imperfect CSI scenario with $T_d = 1$, $f = 2$ GHz and $BW = 0.02f$.

gated in Figure 3. In this figure, we set $\Delta t = \tau_d = 1.9$ msec, implying that the pilots are transmitted every 1.9 msec, and they are received at the satellite with a delay of $T_d = 1$ time step. In this set of figures, we also analyze the effect of satellite handovers on the system in a time span of 300 seconds. The handover is performed based on Algorithm 1, with ϵ taking one of the following three values $\epsilon \in \{0, 0.1, 0.2\}$. First, we show the distance of the satellite that is connected to the users from the centre of the coverage area throughout time. In this figure, the handovers are shown with dotted black lines for $\epsilon = 0$, blue dotted lines for $\epsilon = 0.1$ and red dotted lines for $\epsilon = 0.2$. In the case of $\epsilon = 0$, where the nearest satellite is connected to the users, a total of 6 handovers take place in 6 minutes. When we increase ϵ to 0.1, the number of handovers declines to 4, while the average distance of the connected satellite is only increased moderately. However, in the case of $\epsilon = 0.2$, the connected satellite moves far away from the coverage area before a handover happens. This clearly impacts the quality of the channel, as depicted here. For instance, the minimum absolute value for the channel of the user 1 in the case of $\epsilon = 0.2$ is on the order of 10^{-8} while it is on the order of 10^{-6} for the case of $\epsilon = 0$. Here we also show the performance of the system in terms of average episodic rate, which shows that despite the learning process of the agent, the achievable rate decreases before handovers, due to the low quality of channels. This effect is bolder in the case of $\epsilon = 0.2$. Figure 3 depicts the results under the perfect CSI assumption as an upper bound for realistic cases with delayed CSI. Based on this figure, we set $\epsilon = 0.1$ for the remainder of our simulations to ensure a certain sum rate while keeping the number of handovers relatively low (4 handovers per 6 minutes).

In order to further examine the convergence of the algorithm and the performance of the system in a low pilot rate scenario and $\epsilon = 0.1$ in the handover scheme, under the imperfect CSI assumption, we present Figure 4. Here, we set $\Delta t = \tau_d$, and we compare two scenarios: perfect and imperfect CSI associated with a CSI delay of $T_d = 1$ time step. The performance gap between the perfect CSI assumption and the realistic scenario with delayed CSI is minimal. This is because the agent receives prompt feedback on its actions after a single time step. Additionally, the actor and critic losses in the imperfect CSI case converge almost as quickly as in the perfect CSI scenario. Comparing the results of Figure 4 to those of Figure 2 offers valuable insights. In Figure 2, the pilot rate is very high, resulting in highly correlated estimated channels over time. By contrast, Figure 4 features a lower pilot rate, leading to less correlated CSI over time. One might expect the imperfect CSI case of Figure 4 to perform worse than in Figure 2, but this is not the case. The performances remain nearly identical because, although the channel correlations are lower in Figure 4, they are not excessively low. Hence, the agent does not lose a significant amount of information. In other words, in the high pilot rate scenario, the agent receives redundant information. Furthermore, the imperfect CSI scenario of Figure 4 experiences only a $T_d = 1$ time step delay, compared to the $T_d = 16$ time steps delay of Figure 2. This shorter delay allows the agent in a lower pilot rate environment to converge faster. It is important to note that the lower pilot rate scenario imposes a limitation on the packet transmission rates in both the uplink and the downlink, implying that data packets can be transmitted once every 1.9 milliseconds, but not more frequently. Therefore, each scenario has its own advantages and disadvantages, and the choice between them should be based on the specific application requirements.

All figures presented so far illustrate system performance at an operating frequency of 2 GHz. However, it is important to note that as the frequency increases, the CSI variations become more rapid, leading to a reduced coherence time. Consequently, the correlation between consecutive CSI observations diminishes at a fixed pilot rate, potentially affecting the accuracy of channel estimation and system performance. Thus, Figure 5 compares the system performance at two different operating frequencies, $f = 2$ GHz and $f = 5$ GHz, while maintaining a proportional bandwidth of $BW = 0.02f$ in both cases. Here, we used the handover of $\epsilon = 0.1$ with the pilots being sent every τ_d second. At the higher frequencies, the path loss tends to be higher, which reduces the overall achievable sum rate. Furthermore, for a wider bandwidth, the power of noise also increases, which further reduces the rate. A higher frequency also means that the changes of the channels are faster, which makes it more difficult for the agent to predict appropriate actions based on the delayed channels. In other words, the gap between the perfect and the imperfect CSI scenarios is larger at higher frequencies, which is due to the lower correlation between channels throughout time. The performance discussed here is measured in bit/sec/Hz. When considering the actual data rates in bit/sec, different results emerge. For instance, at the frequency of 5 GHz with a bandwidth of 100 MHz, a sum rate of 600 Mbit/sec can be achieved in the scenario of imperfect CSI. By contrast, at the frequency of 2 GHz with a bandwidth of 40 MHz, the sum rate achieved is 350 Mbit/sec.

In order to compare our results with another relevant study in the literature addressing the delayed CSI problem, Figure 6 is presented. We have chosen the works in [6], [13], [31] as a benchmark for our research. In [6], the channels were predicted using a convolutional neural network (CNN), and a separate CNN was then employed for mapping the predicted channels to appropriate beamforming vectors. Here, 28% of our dataset of channels is used to train the channel prediction CNN, and 10% of the remaining dataset, combined with zero forcing beamforming (ZF), is utilized to train the CNN for beamforming design. Our dataset contains the channels of the selected satellite to $K = 2$ users over 200000 time steps. The comparison demonstrates that the DRL-based method is more effective in handling delayed CSI compared to the supervised learning techniques employed in [6].

We have also implemented the DL-based channel prediction technique from [13], where the current CSI is estimated based on delayed CSI observations. However, since this work does not include a precoding design, we used our own DRL-based method to map the predicted channels to a precoding

matrix. This ensures a fair and meaningful comparison with our robust precoding approach, which directly maps delayed CSI to precoding matrices. As shown in the figure, during the initial episodes, the method in [13] achieves better performance because our DRL-based method has not yet learned the CSI correlations throughout time. However, over time, our approach outperforms the technique in [13], as their method suffers from a constant CSI prediction error. Furthermore, we have compared our results with [31], which employs a non-DL approach to mitigate the effects of delayed CSI. As previously discussed, this technique is designed for frequencies up to 1 GHz and is primarily focused on minimizing MSE rather than maximizing sum rate, explaining its lower average sum rate performance.

Up to here, for all the figures, the reward function in (12) was used. Now, to ensure user fairness in in this algorithm, which is opportunistic by nature, we have applied the reward function (13). The parameters of (14) are: $\zeta_1 = -0.2$, $\zeta_2 = -0.4$, $\zeta_3 = -0.7$, $\zeta_4 = -1$, $\nu_1 = 0.2$, $\nu_2 = 0.15$, $\nu_3 = 0.1$, and $\nu_4 = 0.05$. Based on these chosen parameters, we demonstrated the average achievable rates of each user individually in Figure 7 under an imperfect CSI scenario with $T_d = 1$. As shown here, the opportunistic algorithm is still trying to maximize the rate of one user; however, it ensures a certain quality of service for the other.

D. Computational complexity

In order to compare the computational complexity of our work with the abovementioned studies in the literature, we first compute the time complexity of our DRL-based technique based on [37]. The time complexity of the proposed algorithm depends on the complexity of the neural network operations involved, including the number of layers and the size of the nodes. Both our actor and critic networks consist of multiple dense layers, with each layer having the complexity of $\mathcal{O}(S_{in}S_{out})$ according to [38]. Here, S_{in} and S_{out} denote the input and output size of the respective dense layer, respectively. Thus, given the information in Tables III and IV, the computational complexity of the actor and critic networks are $\mathcal{O}\left(\text{Num. States} \times \text{Num. Actions} + 4(\text{Num. Actions})^2\right)$ and $\mathcal{O}\left(2(\text{Num. States} \times \text{Num. Actions}) + 17.77(\text{Num. Actions})^2\right)$, respectively. Given Num. States $= 2(T_d + 2)MK$ and Num. Actions $= 2MK$, the order of time complexity for our method is scaled by $\mathcal{O}\left((T_d+1)M^2K^2\right)$ for each time step. This indicates that computational complexity is influenced not only by the number of users and antennas but also by the pilot rate. Therefore, a system with a lower pilot rate (e.g. $T_d = 1$) would be the most efficient choice.

VI. Conclusions

We considered the downlink of a multi-antenna satellite system supporting multiple users on the ground. We aimed to construct a TPC matrix on the satellite side to increase the throughput. To do so, precise CSI is needed; however, the CSI available at the satellite is inevitably outdated due to the high propagation delay. To address the delayed CSI challenge, we employed an online RL approach, which exploits the CSI correlations throughout time to construct the TPC matrix. The model converges after approximately 27000 CSI observations, corresponding to 50 seconds with a pilot transmission interval of 1.9 msec. Once trained, the model remains adaptive to environmental changes, ensuring robust service even during handovers or user reassignments, unlike the family of supervised learning methods that require retraining. Our numerical results confirmed the robustness of the proposed method to the CSI imperfections along with different handover strategies and different frequency bands.

Appendix A
Calculation of the Achievable Rate

Let us consider the received signal of the kth user, presented in (1). This equation can be rewritten as

$$\begin{aligned} y^{[k]}(t,f) &= \sum_{i=1}^{K} \mathbf{h}^{[k]}(t,f)^H \mathbf{v}^{[i]}(t,f) s^{[i]}(t,f) + n^{[k]}(t,f) \\ &= \sum_{i=1}^{K} \mathbf{h}^{[k]}(t,f)^H \mathbf{t}^{[i]}(t,f) + n^{[k]}(t,f), \end{aligned} \tag{19}$$

where we have $\mathbf{t}^{[i]}(t,f) = \mathbf{v}^{[i]}(t,f)s^{[i]}(t,f)$. Now, let $\mathcal{I}\left(y^{[k]}(t,f); \mathbf{t}^{[k]}(t,f) | \mathbf{h}^{[k]}(t,f)\right)$ denote the mutual information of the kth user conditioned on the estimated channel. By expanding the mutual information in terms of the differential entropies, we have

$$\begin{aligned} &\mathcal{I}\left(y^{[k]}(t,f); \mathbf{t}^{[k]}(t,f) | \mathbf{h}^{[k]}(t,f)\right) \\ &= \mathcal{H}\left(\mathbf{t}^{[k]}(t,f) | \mathbf{h}^{[k]}(t,f)\right) \\ &\quad - \mathcal{H}\left(\mathbf{t}^{[k]}(t,f) | y^{[k]}(t,f), \mathbf{h}^{[k]}(t,f)\right). \end{aligned} \tag{20}$$

Considering the Gaussian distribution for the transmitted symbols, the first term on the right-hand side of (20) can be simplified to

$$\mathcal{H}\left(\mathbf{t}^{[k]}(t,f) | \mathbf{h}^{[k]}(t,f)\right) = \log\det\left(2\pi e \mathbf{F}^{[k]}(t,f)\right), \tag{21}$$

where we have $\mathbf{F}^{[k]}(t,f) = \mathbb{E}\{\mathbf{t}^{[k]}(t,f)\mathbf{t}^{[k]H}(t,f)\}$ [39]. The second term of (20) is simplified based on the method given in [40]. Based on this method, the second term entropy is upper bound by

$$\begin{aligned} &\mathcal{H}\left(\mathbf{t}^{[k]}(t,f) | y^{[k]}(t,f), \mathbf{h}^{[k]}(t,f)\right) \\ &\leq \mathcal{H}\left(\mathbf{t}^{[k]}(t,f) - \mathbf{g}^{[k]}(t,f) y^{[k]}(t,f) | \mathbf{h}^{[k]}(t,f)\right) \\ &\leq \log\det\left(2\pi e \mathbb{E}\left\{\left|\mathbf{t}^{[k]}(t,f) - \mathbf{g}^{[k]}(t,f) y^{[k]}(t,f)\right|^2\right\}\right). \end{aligned} \tag{22}$$

In (22), $\mathbf{g}^{[k]}(t,f)$ is picked so that $\mathbf{g}^{[k]}(t,f)y^{[k]}(t,f)$ is the linear minimum mean-square error (LMMSE) estimate of

$\mathbf{t}^{[k]}(t,f)$, as demonstrated by

$$\mathbf{g}^{[k]}(t,f) = \frac{\mathbb{E}\left\{y^{[k]}(t,f)\mathbf{t}^{[k]H}(t,f)\right\}}{\mathbb{E}\left\{y^{[k]}(t,f)y^{[k]*}(t,f)\right\}} = \frac{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)+\Gamma^{[k]}(t,f)}, \tag{23}$$

where $\Gamma^{[k]}(t,f) = \sum_{i\neq k}^{K}\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[i]}(t,f)\mathbf{h}^{[k]}(t,f)+\sigma^2$. Thus, the second term in (20) can be upper bound by (24) at the top of the next page.

Now, inspired by [29], we employ the Woodbury matrix identity, as

$$(\mathbf{A}+\mathbf{BCD})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1}. \tag{25}$$

Upon assuming $\mathbf{A}=\mathbf{I}$, $\mathbf{B}=\mathbf{h}^{[k]}(t,f)$, $\mathbf{C}=\left(\Gamma^{[k]}(t,f)\right)^{-1}$ and $\mathbf{D}=\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)$, and using (25), we have

$$\mathbf{I}-\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)+\Gamma^{[k]}(t,f)} = \left(\mathbf{I}+\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\Gamma^{[k]}(t,f)}\right)^{-1}. \tag{26}$$

Now, (24) can be rewritten by (27) at the top of the next page. By substituting (21) and (27) into (20), we have

$$\begin{aligned}
&\mathcal{I}\left(y^{[k]}(t,f);\mathbf{t}^{[k]}(t,f)|\mathbf{h}^{[k]}(t,f)\right)\\
&\geq \log\det\left(\mathbf{I}+\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\Gamma^{[k]}(t,f)}\right)\\
&\overset{(i)}{=} \log\left(1+\frac{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)}{\Gamma^{[k]}(t,f)}\right)\\
&\overset{(ii)}{=} \log\left(1+\frac{|\mathbf{h}^{[k]H}(t,f)\mathbf{v}^{[k]}(t,f)|^2}{\sum_{i\neq k}^{K}|\mathbf{h}^{[k]H}(t,f)\mathbf{v}^{[i]}(t,f)|^2+\sigma^2}\right).
\end{aligned} \tag{28}$$

In (28), the equality (i) results from Sylvester's determinant theorem, i. e., $\det(\mathbf{I}+\mathbf{AB})=\det(\mathbf{I}+\mathbf{BA})$, and the equality (ii) is based on the assumption of $\mathbb{E}\left\{\left|s^{[k]}\right|^2\right\}=1$. Thus, the achievable rate for the kth user can be expressed by the final equation in (28).

REFERENCES

[1] L. Zhen, A. K. Bashir, K. Yu, Y. D. Al-Otaibi, C. H. Foh, and P. Xiao, "Energy-Efficient Random Access for LEO Satellite-Assisted 6G Internet of Remote Things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5114–5128, 2021.

[2] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite Communications Supporting Internet of Remote Things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, 2016.

[3] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge, and C.-X. Wang, "5G Embraces Satellites for 6G Ubiquitous IoT: Basic Models for Integrated Satellite Terrestrial Networks," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 399–14 417, 2021.

[4] C.-H. Lin, S.-C. Lin, and L. C. Chu, "A Low-Overhead Dynamic Formation Method for LEO Satellite Swarm Using Imperfect CSI," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 6923–6936, 2024.

[5] Y. Omid, Z. M. Bakhsh, F. Kayhan, Y. Ma, and R. Tafazolli, "Space MIMO: Direct Unmodified Handheld to Multi-Satellite Communication," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 1447–1452.

[6] Y. Zhang, A. Liu, P. Li, and S. Jiang, "Deep Learning (DL)-Based Channel Prediction and Hybrid Beamforming for LEO Satellite Massive MIMO System," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 705–23 715, 2022.

[7] T. Van Chien, E. Lagunas, T. M. Hoang, S. Chatzinotas, B. Ottersten, and L. Hanzo, "Space-Terrestrial Cooperation Over Spatially Correlated Channels Relying on Imperfect Channel Estimates: Uplink Performance Analysis and Optimization," *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 773–791, 2023.

[8] D. Tuzi, E. F. Aguilar, T. Delamotte, G. Karabulut-Kurt, and A. Knopp, "Distributed Approach to Satellite Direct-to-Cell Connectivity in 6G Non-Terrestrial Networks," *IEEE Wireless Communications*, vol. 30, no. 6, pp. 28–34, 2023.

[9] M. Hosseinian, J. P. Choi, S.-H. Chang, and J. Lee, "Review of 5G NTN Standards Development and Technical Challenges for Satellite Integration With the 5G Network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 8, pp. 22–31, 2021.

[10] M. Khalid, J. Ali, and B.-h. Roh, "Artificial Intelligence and Machine Learning Technologies for Integration of Terrestrial in Non-Terrestrial Networks," *IEEE Internet of Things Magazine*, vol. 7, no. 1, pp. 28–33, 2024.

[11] Q. Gao, M. Jia, Q. Guo, X. Gu, and L. Hanzo, "Jointly Optimized Beamforming and Power Allocation for Full-Duplex Cell-Free NOMA in Space-Ground Integrated Networks," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2816–2830, 2023.

[12] S. Mahboob and L. Liu, "Revolutionizing Future Connectivity: A Contemporary Survey on AI-Empowered Satellite-Based Non-Terrestrial Networks in 6G," *IEEE Communications Surveys and Tutorials*, vol. 26, no. 2, pp. 1279–1321, 2024.

[13] Y. Zhang, Y. Wu, A. Liu, X. Xia, T. Pan, and X. Liu, "Deep Learning-Based Channel Prediction for LEO Satellite Massive MIMO Communication System," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1835–1839, 2021.

[14] L. You, K.-X. Li, J. Wang, X. Gao, X.-G. Xia, and B. Ottersten, "Massive MIMO Transmission for LEO Satellite Communications," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1851–1865, 2020.

[15] K.-X. Li, L. You, J. Wang, X. Gao, C. G. Tsinos, S. Chatzinotas, and B. Ottersten, "Downlink Transmit Design for Massive MIMO LEO Satellite Communications," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1014–1028, 2022.

[16] M. Jalali, E. Lagunas, A. Haqiqatnejad, S. Kisseleff, and S. Chatzinotas, "Downlink Beamforming Strategies for Interference-Aware NGSO Satellite Systems," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3468–3483, 2024.

[17] M. Y. Abdelsadek, G. Karabulut-Kurt, H. Yanikomeroglu, P. Hu, G. Lamontagne, and K. Ahmed, "Broadband Connectivity for Handheld Devices via LEO Satellites: Is Distributed Massive MIMO the Answer?" *IEEE Open Journal of the Communications Society*, vol. 4, pp. 713–726, 2023.

[18] W. Jiang and H. D. Schotten, "Neural Network-Based Fading Channel Prediction: A Comprehensive Overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, 2019.

[19] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine Learning-Based Channel Prediction in Massive MIMO With Channel Aging," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2960–2973, 2020.

[20] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "DeepChannel: Wireless Channel Quality Prediction Using Deep Learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 443–456, 2020.

[21] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse Channel Estimation and Hybrid Precoding Using Deep Learning for Millimeter Wave Massive MIMO," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2838–2849, 2020.

[22] M. J. Kang, J. H. Lee, and S. H. Chae, "Channel Estimation with DnCNN in Massive MISO LEO Satellite Systems," in *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2023, pp. 825–827.

[23] T. X. Vu, S. Bhandari, M. Minardi, H. Van Nguyen, and S. Chatzinotas, "3GPP New Radio Precoding in NGSO Satellites: Channel Prediction and Dynamic Resource Allocation," in *2023 IEEE Statistical Signal Processing Workshop (SSP)*, 2023, pp. 115–119.

[24] Z. Geng, C. She, D. Zhang, C. Li, Y. Li, and B. Vucetic, "Zero-Shot Recurrent Graph Neural Networks for Beam Prediction in Non-Terrestrial Networks," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 1400–1405.

$$\mathcal{H}\left(\mathbf{t}^{[k]}(t,f)|y^{[k]}(t,f),\mathbf{h}^{[k]}(t,f)\right) \leq \log\det\left(2\pi e\left(\mathbf{F}^{[k]}(t,f)-\frac{\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)+\Gamma^{[k]}(t,f)}\right)\right). \tag{24}$$

$$\begin{aligned}\mathcal{H}\left(\mathbf{t}^{[k]}(t,f)|y^{[k]}(t,f),\mathbf{h}^{[k]}(t,f)\right) &\leq \log\det\left(2\pi e\mathbf{F}^{[k]}(t,f)\left(\mathbf{I}-\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)\mathbf{h}^{[k]}(t,f)+\Gamma^{[k]}(t,f)}\right)\right)\\ &= \log\det\left(2\pi e\mathbf{F}^{[k]}(t,f)\left(\mathbf{I}+\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\Gamma^{[k]}(t,f)}\right)^{-1}\right)\\ &= \log\det\left(2\pi e\mathbf{F}^{[k]}(t,f)\right)-\log\det\left(\mathbf{I}+\frac{\mathbf{h}^{[k]}(t,f)\mathbf{h}^{[k]H}(t,f)\mathbf{F}^{[k]}(t,f)}{\Gamma^{[k]}(t,f)}\right).\end{aligned} \tag{27}$$

[25] V. N. Ha, Z. Abdullah, G. Eappen, J. C. M. Duncan, R. Palisetty, J. L. G. Rios, W. A. Martins, H.-F. Chou, J. A. Vasquez, L. M. Garces-Socarras, H. Chaker, and S. Chatzinotas, "Joint Linear Precoding and DFT Beamforming Design for Massive MIMO Satellite Communication," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 1121–1126.

[26] F. Wang, D. Jiang, Z. Wang, J. Chen, and T. Q. S. Quek, "Seamless Handover in LEO Based Non-Terrestrial Networks: Service Continuity and Optimization," *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 1008–1023, 2023.

[27] B. Zheng, S. Lin, and R. Zhang, "Intelligent Reflecting Surface-Aided LEO Satellite Communication: Cooperative Passive Beamforming and Distributed Channel Estimation," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 10, pp. 3057–3070, 2022.

[28] E. T. Michailidis, P. Theofilakos, and A. G. Kanatas, "Three-Dimensional Modeling and Simulation of MIMO Mobile-to-Mobile via Stratospheric Relay Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2014–2030, 2013.

[29] Y. Omid, S. M. Mahdi Shahabi, C. Pan, Y. Deng, and A. Nallanathan, "Robust Beamforming Design for an IRS-Aided NOMA Communication System with CSI Uncertainty," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[30] Y. Omid, S. M. Shahabi, C. Pan, Y. Deng, and A. Nallanathan, "Low-Complexity Robust Beamforming Design for IRS-Aided MISO Systems With Imperfect Channels," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1697–1701, 2021.

[31] Y. Omid, S. Lambotharan, and M. Derakhshani, "Tackling Delayed CSI in a Distributed Multi-Satellite MIMO Communication System," in *2024 19th International Symposium on Wireless Communication Systems (ISWCS)*, 2024, pp. 1–6.

[32] S. Nath, M. Baranwal, and H. Khadilkar, "Revisiting State Augmentation Methods for Reinforcement Learning with Stochastic Delays," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1346–1355.

[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, feb 2015.

[34] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 1057–1063, 2000.

[35] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," *[Online]. Available: arXiv:1910.02398*, 2015.

[36] E. Altman and P. Nain, "Closed-Loop Control with Delayed Information," *ACM sigmetrics performance evaluation review*, vol. 20, no. 1, pp. 193–204, 1992.

[37] W. Zhang, M. Derakhshani, G. Zheng, and S. Lambotharan, "Constrained Risk-Sensitive Deep Reinforcement Learning for eMBB-URLLC Joint Scheduling," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 10 608–10 624, 2024.

[38] Y. Yang, F. Gao, C. Xing, J. An, and A. Alkhateeb, "Deep Multimodal Learning: Merging Sensory Data for Massive MIMO Channel Prediction," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1885–1898, 2021.

[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications. New York, USA: John Wiley & Sons, Inc., 1991.

[40] M. Medard, "The Effect Upon Channel Capacity in Wireless Communications of Perfect and Imperfect Knowledge of the Channel," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 933–946, 2000.