Abstract—Unmanned Aerial Vehicle (UAV)-assisted communication has drawn increasing attention recently. In this paper, we investigate 3D UAV trajectory design and band allocation problem considering both the UAV’s energy consumption and the fairness among the ground users (GUs). Specifically, we first formulate the energy consumption model of a quad-rotor UAV as a function of the UAV’s 3D movement. Then, based on the fairness and the total throughput, the fair throughput is defined and maximized within limited energy. We propose a deep reinforcement learning (DRL)-based algorithm, named as EEFC-TDBA (energy-efficient fair communication through trajectory design and band allocation) that chooses the state-of-the-art DRL algorithm, deep deterministic policy gradient (DDPG), as its basis. EEFC-TDBA allows the UAV to: 1) adjust the flight speed and direction so as to enhance the energy efficiency and reach the destination before the energy is exhausted; and 2) allocate frequency band to achieve fair communication service. Simulation results are provided to demonstrate that EEFC-TDBA outperforms the baseline methods in terms of the fairness, the total throughput, as well as the minimum throughput.

Index Terms—3D UAV trajectory, band allocation, energy-efficient, fair communication, deep reinforcement learning.

I. INTRODUCTION

UNMANNED Aerial Vehicle (UAV)-assisted communication has attracted increasing attention recently for its providing flexible and cost-effective communication service as well as enhancing coverage under various scenarios [1], [2]. Due to the high mobility, the UAV-assisted communication systems are widely used for emergency communications and broadband connectivity at remote areas [3], [4]. For example, when the terrestrial communication infrastructures are severely damaged by catastrophic natural disasters [5], [6], UAVs are capable of serving as temporary base stations (BSs) and can connect users to the backbone network. Besides, UAVs are likely to provide better communications due to the higher chance of line-of-sight (LoS) links to ground users (GUs), compared to traditional terrestrial communication systems.

Existing researches on UAV-assisted communications can mainly be categorised into two groups: 1) UAVs are used for enlarging the communication coverage [7]–[11] and act as mobile BSs to serve GUs; and 2) UAVs are dispatched as data relay to connect two or more distant users or user groups [12]–[16], and can provide low latency service [17]. In [18], the authors leverage better link qualities from the line-of-sight (LoS) links of UAV communication systems to enable low latency communications. In [19], the authors design a real-time data delivery feature for oneM2M standard platform. In [20], the authors design a 3D trajectory planning and scheduling algorithm to minimize the average pathloss. In [21], the authors apply penalty dual-decomposition (PDD) technique to maximize the total throughput of UAV communications with orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA) modes, which is a nonconvex optimization problem. In [22], Al-Hourani et al. present an analytical approach to optimize the altitude of low-altitude aerial platforms (LAPs) in order to provide maximum communication coverage. Lyu et al. [23] propose a new cyclical multiple access (CMA) scheme to allocate communication time between UAV and GUs in order to maximize their minimum throughput. Zhang et al. [24] find a closed-form low-complexity solution with joint trajectory design and power control to minimize the outage probability of a UAV relay network.

Despite the advantages, such as high mobility, flexible deployment, and low operational costs, UAV-assisted communication systems face the energy constraint challenge. For example, due to the size and weight constraints, the on-board
energy of UAV is limited, which leads to endurance and performance degradation. As a result, the energy efficiency, defined as the information bits per unit energy consumption, is a key issue in UAV-assisted communications. In [25], the authors derive a mathematical model on the propulsion energy consumption of fixed-wing UAVs and then maximize the energy-efficiency through UAV trajectory design. Zeng et al. [26], [27] obtain a closed-form propulsion power consumption model for rotary-wing UAVs, and then optimize UAV trajectory and time allocation among GUs to minimize the total energy consumption during the flight. All these existing problems can be summarized as follows:

- The energy consumption models in [25]–[27] are all restricted to horizontal flight, while none of them consider UAV’s 3D trajectory.
- Most works transform the original nonconvex problem into a convex one and then apply the convex optimization toolbox such as CVX [28] to solve the problem. Actually, such approach converts the continuous trajectory design into the fly-hover-communicate design, which simplifies the problem at the cost of accuracy.
- The complexities of these algorithms increase rapidly with the number of users, flight time, and the number of iterations. Besides, the traditional optimization methods cannot deal with users’ movement, i.e., they are not capable of selecting the UAV trajectory or allocating communication resources according to the users’ current location.

Recently, there have been some researches leveraging deep reinforcement learning (DRL) [29] for UAV-assisted communications. They model the problems as the Markov decision process (MDP) [30], where an agent observes the state of environment, takes an action, and obtains a reward. Then the environment changes into another state. The objective of DRL algorithms is to maximize the expected accumulative reward without the need of transforming nonconvex problem to convex one. DRL can utilize the data process ability of deep neural network (DNN) to deal with the users’ movements. In [32], Yin et al. optimize UAV trajectory to maximize the throughput without considering energy consumption and resource allocation. In [33], K-means is used to obtain the cell partition of the users, and a Q-learning based algorithm is proposed to select the deployment location. In [34], the authors propose a DRL-based energy-efficient control method for fair communication coverage. However, the algorithm only considers the coverage of each UAV, while the objective is purely to enhance the coverage time of each cell rather than the throughput. Besides, [34] does not apply the resource allocation to further improve the performance.

In this paper, we investigate the problems of energy-efficiency and fair communication service for a quad-rotor UAV-assisted system by jointly designing the 3D trajectory of UAV and the frequency band allocation of moving GUs. We propose a DRL based algorithm, named as EEFC-TDBA, to enable the UAV to provide energy-efficient and fair communication service through trajectory design and frequency band allocation. To cope with the continuous control problem with unlimited state and action space, EEFC-TDBA is designed based on deep deterministic policy gradient (DDPG) [35]. The main contributions of this paper are summarized as follows:

- We derive a mathematical model for the propulsion energy-consumption of a quad-rotor UAV in 3D flight.
- We formulate a UAV energy-efficient and fair communication problem where the UAV chooses the flying direction and the band allocated to each GU, and propose a DDPG based algorithm to solve it. When applying DDPG, there exist dimension imbalance, the gradient vanishing, and the training oscillation issues. We next design dimension spread, pre-activation, softmax reference techniques to address the issues respectively.
- The proposed algorithm takes the UAV’s and GUs’ location as well as the destination coordinates into consideration, and hence can tackle the GUs’ movement issue.
- A fairness utility is applied to trade off the throughput maximization and fairness among the GUs.

The rest of this paper is organized as follows. Section II presents the system model and formulates the fair throughput maximization problem. Section III-A provides a brief review of DRL. In Section III, the EEFC-TDBA algorithm is described in detail, while Section IV presents the performance evaluation. Finally, we conclude the paper in Section V.

Notations: In this paper, scalars are denoted by italic letters, and vectors are denoted by boldface letters. The Euclidean norm of a vector is denoted by $\| \cdot \|$, and $\{ \cdot \}$ denotes an array; For a time-dependent function $a(t)$, $\dot{a}(t)$ and $\ddot{a}(t)$ denote the first- and second-order derivatives with respect to $t$. $\mathbb{R}^M$ denotes the space of M-dimensional real vectors; $\mathbb{E}_\pi[\cdot]$ denotes expectation of a random variable following policy $\pi$.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

As shown in Fig. 1, we consider a quad-rotor UAV system where a UAV serves as BS for $K$ GUs, $K = \{ 1, \cdots, K \}$, and has a total energy $E_{\text{max}}$. Denote $E(t)$ as the remaining energy of UAV at time $t$ and denote $T_t$ as the total flight time of UAV, i.e., $E(0) = E_{\text{max}}$ and $E(T_t) = 0$. The GUs move on the ground at constant speeds, and the location of GU $k$
at time $t$ is denoted by $u_k(t) \in \mathbb{R}^3$, $0 \leq t \leq T_i$. Meanwhile, the 3D Cartesian coordinate of the quad-rotor UAV location at time $t$ is denoted by $u(t) = [(x(t), y(t)), z(t)] \in \mathbb{R}^3$, where $(x(t), y(t))$ is the UAV location projected on the horizontal plane, and $z(t)$ is the UAV altitude. The UAV adopts the frequency division multiple access (FDMA) to serve $K$ GUs with a total bandwidth $B$. Let $B_k(t)$ be the communication bandwidth allocated to GU $k$ at time $t$. Then there is

$$
\sum_{k=1}^{K} B_k(t) = B, \forall t \in [0, T_i].
$$

(1)

Due to the higher chance of the line-of-sight (LoS) connectivity [9], the air-to-ground (ATG) channel is different from the terrestrial channel and mainly depends on the elevation angle and the type of the propagation environment. To take the occurrence of LoS links into account, we adopt the probabilistic pathloss model in [22], where the probability of LoS connectivity between the UAV and GU $k$ at time $t$ is

$$
h_{k,\text{LoS}}^L(t) = \frac{1}{1 + \eta_a \exp\left(-\eta_b \left(\arcsin\left(\frac{z(t)}{dz(t)}\right) - \eta_c\right)\right)}.
$$

(2)

Here $\eta_a$ and $\eta_b$ are constants related to the type of the propagation environment, and $d_k(t) = ||u(t) - u_k(t)||$ denotes the time-varying distance from the UAV to GU $k$. It can be inferred that the corresponding probability of the non-LoS (NLoS) links is $h_{k,\text{NLoS}}^N(t) = 1 - h_{k,\text{LoS}}^L(t)$.

The pathloss expressions of LoS and NLoS links between the UAV to and GU $k$ are

$$
L_{k,\text{LoS}}(t) = L_{k,\text{FS}}(t) + \eta_{\text{LoS}},
$$

(3)

$$
L_{k,\text{NLoS}}(t) = L_{k,\text{FS}}(t) + \eta_{\text{NLoS}},
$$

(4)

where $L_{k,\text{FS}}(t) = 20 \log d_k(t) + 20 \log f_c + 20 \log \left(\frac{20}{\lambda t}\right)$ is the free space pathloss, $f_c$ denotes the carrier frequency, and $v_c$ represents the velocity of light. Besides, $\eta_{\text{LoS}}$ and $\eta_{\text{NLoS}}$ are the excessive pathloss [36] for LoS and NLoS links, respectively. Then the pathloss expression between the UAV and GU $k$ becomes

$$
L_k(t) = h_{k,\text{LoS}}^L(t) \times L_{k,\text{LoS}}(t) + h_{k,\text{NLoS}}^N(t) \times L_{k,\text{NLoS}}(t)
$$

$$
= L_{k,\text{FS}}(t) + h_{k,\text{LoS}}^L(t)\eta_{\text{LoS}} + (1 - h_{k,\text{LoS}}^L(t))\eta_{\text{NLoS}}.
$$

(5)

The transmission rate between the UAV and GU $k$ can be expressed as

$$
R_k(t) = B_k(t) \log_2 \left(1 + \frac{P_c}{\beta_k(t)n_0B_k(t)}\right),
$$

(6)

where $P_c$ is the communication power for each GU $k$, $n_0$ is the noise power spectral density, and $\beta_k(t) = 10^{L_k(t)/10}$. The power allocated to each GU is equal. The overall capacity of the UAV is

$$
R_c(t) = \sum_{k=1}^{K} R_k(t).
$$

(7)

Note that given the trajectories $\{u_k(t)\}_{k \in K}$ of all GUs, $R_c(t)$ is a function of the UAV location $\{u(t)\}$ and the frequency band allocation $\{B_k(t)\}$. Thus, the total UAV throughput before time $t$ is

$$
\tilde{R}(t) = \tilde{R}(\{u(t)\}, \{B_k(t)\}) = \int_{0}^{t} R_c(t) dt,
$$

(8)

and the throughput for each GU is

$$
\tilde{R}_k(t) = \tilde{R}_k(\{u(t)\}, \{B_k(t)\}) = \int_{0}^{t} R_k(t) dt.
$$

(9)

B. Energy Consumption Model for Quad-Rotor UAV

The energy consumption of the UAV consists of two parts: one is for the communication, and the other is the propulsion energy for generating thrusts$^3$ to help the UAV overcome the drag$^4$ and gravity. In practice, the energy for communication is often much smaller than that for flight by two orders of magnitude [25]. Therefore, the energy consumption of communication is ignored in this paper.

For better exposition, we only consider the acceleration that is in a straight line with velocity while omit the acceleration component that is perpendicular to velocity. Thus the UAV can change direction instantaneously without extra energy consumption, which is reasonable because the quad-rotor UAV can easily steer by adjusting the rotation rate of four rotors. As derived in Appendix A, the thrust of each rotor is a function of UAV velocity $v$ and acceleration $a_c$. The velocity direction vector is denoted by $v_d$, and $v = ||v||$ is the UAV speed. Then the thrust of each rotor can be expressed as

$$
T_h(v, a_c) = \frac{1}{n_r} \left( m ||a_c|| + \frac{1}{2} n_r S_{FP} v_d - mg \right),
$$

(10)

where $n_r$ is the number of rotors; $m$ denotes the UAV mass; $\rho$ and $S_{FP}$ are the air density and fuselage equivalent flat plate area; $g$ is the gravity acceleration vector. Then the propulsion power can be expressed as (11), shown at the bottom of the page, where $\delta$ is the local blade section drag coefficient; $c_T$ denotes the thrust coefficient based on disc area; $A$ and $c_s$ are the disc area for each rotor and rotor solidity respectively; $c_f$ is the incremental correction factor of induced power; $\tau_c$

$^3$The thrusts are the forces produced by the rotors and move the UAV forward.

$^4$The drag is the aerodynamic force component that is in the opposite direction of the UAV’s motion.

$$
P(v, T_h) = n_r \left[ \delta \left( \frac{T_h}{c_T \rho A} + 3v^2 \right) \sqrt{\frac{T_h \rho c_s^2 A}{c_T}} + (1 + c_f)T_h \left( \sqrt{\frac{T_h^2}{4 \rho A^2} + \frac{v^4}{4} - \frac{v^2}{2}} \right)^2 \frac{mgv}{n_r} \sin \tau_c + \frac{1}{2} d_0 v^3 \rho c_s A \right]
$$

(11)
denotes the climb angle; \( d_0 \) is the fuselage drag ratio for each rotor. The detailed introduction of the parameters is in listed in Appendix A.

Remark 1: Considering \( ||\mathbf{a}_c|| = 0 \) and the UAV flies in the horizontal plane, i.e., \( \tau_c = 0 \), we plot the propulsion power versus the UAV velocity in Fig. 2. It is seen that the propulsion power consumption does not increase monotonously with the UAV velocity. When the UAV flies at about 15 m/s, the power consumption achieves the minimum. Therefore, the UAV would be able to fly for the longest time with the fixed energy, which may lead to the maximum total throughput.

Considering \( v(t) \triangleq \dot{u}(t) \) and \( \mathbf{a}_c(t) \triangleq \ddot{u}(t) \), the propulsion power is essentially a function of the UAV trajectory \( \{u(t)\} \). Then with a given trajectory \( \{u(t)\} \), then the remaining energy of the UAV can be expressed as

\[
E(t) = E_{\text{max}} - \int_0^t P(\dot{u}(\tau), T_h(\dot{u}(\tau), \ddot{u}(\tau))) \, d\tau.
\]

**C. Problem Formulation**

The UAV energy efficiency of the UAV-assisted communication system can be defined as \( R(T_i)/E_{\text{max}} \). Since the UAV battery capacity \( E_{\text{max}} \) is a constant, maximizing the energy efficiency is equivalent to maximizing the total throughput before the UAV battery runs out.

However, maximizing the total throughput may lead to unfairness problem, where the UAV may tend to hover close to some GUs, while the other GUS suffer from small throughput all the time. To tackle this issue, we refer to [34] and define the throughput ratio of GU \( k \) as:

\[
f_k(t) = \frac{\bar{R}_k(t)}{R(t)}.
\]

Then we apply Jain’s fairness index [37] to measure the fairness among GUs:

\[
\bar{f}(t) = \frac{(\sum_{k=1}^K f_k(t))^2}{K(\sum_{k=1}^K f_k(t))^2}.
\]

Obviously, \( \bar{f}(t) \in [0, 1] \) holds. The smaller the differences among the throughput ratios \( \{f_k\}_{k \in K} \) are, the greater fair index \( \bar{f} \) is. As a result, larger fair index means fairer communication service. Note that both the fair index and the total throughput of UAV are the functions of the UAV trajectory \( \{u(t)\}_{t \in [0, T_i]} \) and the frequency band allocation \( \{B_k(t)\}_{t \in [0, T_i]} \). Then we may define the total fair throughput during the whole mission as

\[
\bar{R}_f(\{u(t)\}, \{B_k(t)\}) = \int_0^{T_i} \bar{f}(\tau) R_c(\tau) \, d\tau.
\]

In order to balance the total throughput and fairness, our objective is to maximize the fair throughput by designing the UAV trajectory \( \{u(t)\}_{t \in [0, T_i]} \) and allocating the frequency band \( \{B_k(t)\}_{t \in [0, T_i]} \). The problem can be mathematically formulated as

\[
(P1) : \max_{\{u(t)\}, \{B_k(t)\}} \bar{R}_f(\{u(t)\}, \{B_k(t)\})
\]

s.t. \( E(0) = E_{\text{max}}, \ E(T_i) = 0, \)

\( u(0) = u_0, \ u(T_i) = u_c, \)

\( v(0) = 0, \)

\( B_k(t) \geq B_{\text{min}}, \ \forall t \in [0, T_i], \)

\( \sum_{k=1}^K B_k(t) = B(t), \ \forall t \in [0, T_i], \)

\( 0 \leq v(t) \leq v_{\text{max}}, \ \forall t \in [0, T_i], \)

\( 0 \leq ||\mathbf{a}_c(t)|| \leq a_{\text{max}}, \ \forall t \in [0, T_i], \)

\( z_{\text{min}} \leq z(t) \leq z_{\text{max}}, \ \forall t \in [0, T_i], \)

where \( u_0, u_c \in \mathbb{R}^3 \) are the initial and final locations of the UAV, \( B_{\text{min}} \) is the minimum bandwidth to guarantee the basic communication service for each GU, and \( z_{\text{min}} \) and \( z_{\text{max}} \) are the altitude constraints. At \( u_0 \), the UAV takes off with \( v(0) = 0 \) and needs to reach \( u_c \) before it runs out of energy so as to be recharged for the next mission. Note that, problem \( (P1) \) is difficult to solve because the optimization parameters like the UAV trajectory \( \{u(t)\}_{t \in [0, T_i]} \) and the frequency band allocation \( \{B_k(t)\}_{t \in [0, T_i]} \) are continuous time series.

To make problem \( (P1) \) trackable, we divide the time into multiple slots with duration \( \delta_t \). Then the UAV trajectory \( u(t) \) can be characterized by discrete-time UAV location \( u[n] = u(n\delta_t), \ n = 0, 1, \ldots, N_t, \) where \( N_t = [T_i/\delta_t] \). Similarly, we have \( f[n] = f(n\delta_t), \ v[n] = v(n\delta_t), \ a_c[n] = a_c(n\delta_t) \) and \( B_k[n] = B_k(n\delta_t) \). The objective (15) can be rewritten as

\[
\bar{R}_f(\{u[n]\}, \{B_k[n]\}) = \sum_{n=0}^{N_t} \bar{f}(n) R_c(n) \delta_t.
\]

During each time slot, we assume that the acceleration remains constant and the flight direction is fixed. Thus we...
have
\[ v[n + 1] = v[n] + a_c[n] \delta_t, \] (25)
\[ u[n + 1] = u[n] + (v[n] \delta_t + \frac{1}{2} a_c[n] \delta_t^2) \nu_t], \] (26)
where \( a_c = \mathcal{E}_a[a_c]|a_c|, \) and \( \mathcal{E}_a \) is an indicator that if \( \mathcal{E}_a = 1, \) then the direction of acceleration is the same as that of velocity; otherwise, if \( \mathcal{E}_a = -1, \) then the acceleration and velocity are in opposite directions. Moreover, the remaining energy (12) in time-slotted fashion is re-written as
\[ E(n) = E_{\text{max}} - \sum_{i=0}^{n-1} P(v(i), T_h(i)) \delta_t. \] (27)

According to (25) and (26), \( \{u[n]\}, \{v[n]\} \) and \( \{a_c[n]\} \) are linear with each other. Thus we could optimize the UAV velocity \( \{v[n]\} \) or the UAV acceleration \( \{a_c[n]\} \) instead of directly optimizing the UAV trajectory \( \{u[n]\} \). As a result, the problem (P1) can be re-expressed as
\[ (P2) : \max_{\{u[n]\}, \{a_c[n]\}} \quad \bar{R}_f(\{u[n]\}, \{B_k[n]\}) \]
\[ \text{s.t.} \quad E[0] = E_{\text{max}}, \quad E[N_t] = 0, \] (28)
\[ u[0] = u_0, \quad u[N_t] = u_c, \] (29)
\[ v[0] = 0, \] (30)
\[ B_k[n] \geq B_{\text{min}}, \quad n = 0, 1, \ldots, N_t \] (31)
\[ \sum_{k=1}^{K} B_k[n] = B, \quad n = 0, 1, \ldots, N_t \] (32)
\[ 0 \leq v[n] \leq v_{\text{max}}, \quad n = 0, 1, \ldots, N_t \] (33)
\[ -a_{\text{max}} \leq a_c[n] \leq a_{\text{max}}, \quad n = 0, 1, \ldots, N_t \] (34)
\[ z_{\text{min}} \leq z[n] \leq z_{\text{max}}, \quad n = 0, 1, \ldots, N_t \] (35)

Problem (P2) is a non-convex optimization problem with thousands of optimization variables since they are time-varying. In fact, without considering the complex energy consumption model (11) and frequency band allocation, problem (P2) reduces to a travelling salesman problem (TSP) [38], which is known to be NP-hard. Hence, problem (P2) is too difficult to be solved by traditional optimization methods. Fortunately, DRL can search the solution from a large policy space and can capture the feature of the energy consumption model with the powerful data processing capability.

III. DRL B ASED UAV TRAJECTORY DESIGN AND FREQUENCY BAND ALLOCATION

In this section, we present the EEFC-TDBA algorithm for 3D UAV trajectory design and communication frequency band allocation. In EEFC-TDBA, the UAV is treated as the agent. At each time slot, the UAV observes the state \( s(i) \), inputs it to the network, and outputs the action \( a(i) \). Then the UAV receives reward \( r(i) \), and the state turns into \( s(i + 1) \). The corresponding experience \( (s(i), a(i), r(i), s(i + 1)) \) is stored in a replay buffer for the training of the network.

The DRL approach has two phases: training phase and implementation phase. In the training phase, the DNN is trained offline, and the exploration is needed to search the optimal policy. While in the implementation phase, DNN only forwards propagation, which consumes much less resources than training. Besides, there is no need for exploration in implementation phase.

A. Preliminaries

We first provide a brief introduction for DDPG that is a DRL algorithm within actor-critic framework [39]. In DDPG, the critic \( Q(s, a; \theta^Q) \) evaluates the action-value function under the actor policy \( \pi(s; \theta^\pi) \), where \( \theta^\pi \) and \( \theta^Q \) refer to the parameters of actor and critic networks.

However, a non-linear function approximator, e.g., DNN, is known to be unstable and even cause divergence when applied in DRL. Two techniques are usually used in DRL to resolve this issue: experience replay and target network [40]. DRL samples a mini-batch of experiences from the replay buffer that stores state transition samples collected during learning. The random samples break the correlation between sequential samples and stabilize the training process. Moreover, the target networks of actor and critic \( \pi'(s; \theta^\pi') \) and \( Q'(s, a; \theta^Q') \) are used to compute the update target and have the same architectures as the learned networks \( Q(s, a; \theta^Q) \) and \( \pi(s; \theta^\pi) \). Specifically, the critic network can be trained by minimizing the loss:
\[ L(\theta^Q) = \frac{1}{N_b} \sum_i [y(i) - Q(s(i), a(i); \theta^Q)]^2, \] (36)
where
\[ y(i) = r(i) + \gamma Q \left(s(i + 1), \pi' \left(s(i + 1); \theta^\pi'\right); \theta^Q\right), \] (37)
is the update target, and \( N_b \) is the batch size. In addition, the actor is trained by minimizing the actor loss
\[ L(\theta^\pi) = \frac{1}{N_b} \sum_i -Q(s(i), \pi(s(i); \theta^\pi); \theta^Q). \] (38)

The parameters of the target networks are updated by slowly tracking the learned networks: \( \theta' \leftarrow \epsilon \theta + (1 - \epsilon) \theta' \) with \( \epsilon \ll 1 \) [35].

B. State Space

As mentioned in Section II-A, the throughput is related to the pathloss between the UAV and GUs. However, compared with the pathloss, the locations of the UAV and GUs can be obtained more easily, since most cell phones are equipped with GPS sensors. Besides, the locations of GUs change all the time. As a result, the state includes the GUs’ locations \( \{w_k(n)\}_{k \in K} \) and the UAV location \( u(n) \) such that UAV can deal with movement of GUs. According to Section II-C, the UAV needs to reach the destination \( u_c \) before its energy runs out. Thus the target spot \( u_c \) and the current energy \( E(n) \) should be taken into consideration in the state. Moreover, the state includes the current speed of the UAV \( v(n) \), which aims to remind the UAV not to exceed the acceleration constraint (34).

In summary, the state can be formulated as
\[ s(n) = \{\{w_k(n)\}_{k \in K}, u(n), v(n), u_c, E(n)\}, \] (39)
and has \( 3K + 8 \) dimensions.
The actions for frequency band allocation strategy should satisfy the constraints (31) and (32).

\[ \sum_{\lambda} \text{c} \]

where \( C \) is the frequency band allocation strategy.

![Diagram](image)

Fig. 3. The spherical coordinate.

C. Action Space

The action \( a(n) \) of the UAV trajectory design and frequency band allocation consists of two parts:

- \( v(n+1) \): the UAV velocity at the next time slot with \( \|v(n+1)\| \in [0,v_{\text{max}}] \);
- \( \{B_k(n)\}_{k \in \mathcal{K}} \): the frequency band allocation.

1) UAV Velocity: As shown in Fig. 3, we use the spherical coordinate \( \{v, \varphi_p, \varphi_a\} \) in order to describe the UAV speed and flight direction more conveniently, where \( \varphi_p \) is polar angle from the positive z-axis with \( 0 \leq \varphi_p \leq \pi \), and \( \varphi_a \) is known to be the azimuthal angle in the xy-plane from the x-axis with \( -\pi \leq \varphi_a \leq \pi \).

For convenience, we apply the normalized representation for the UAV velocity:

\[ v = \lambda_v \cdot v_{\text{max}}, \quad \varphi_p = \lambda_{\varphi_p} \cdot \pi, \quad \varphi_a = \lambda_{\varphi_a} \cdot \pi, \]

where \( \lambda_v, \lambda_{\varphi_p}, \lambda_{\varphi_a} \in [0,1] \) and \( \lambda_{\varphi_a} \in [-1,1] \).

2) Frequency Band Allocation: The frequency band allocation strategy should satisfy the constraints (31) and (32). The actions for frequency band allocation \( \{B_k(n)\}_{k \in \mathcal{K}} \) can be re-expressed as the ratios of bandwidth allocated to GU \( k \) to the total bandwidth \( \{\lambda_k^b(n)\}_{k \in \mathcal{K}} \), i.e.,

\[ B_k(n) = \lambda_k^b(n)B, \]

where \( \sum_{k=0}^{K} \lambda_k^b(n) = 1 \) and \( \lambda_k^b(n) \geq \frac{B_{\text{min}}}{B} \).

In summary, the action \( a(n) \) has \( (3+K) \) dimensions, in which "3" refers to the UAV velocity related action, and "K" refers to the frequency band allocation part for K GUs.

D. Reward Design

In DRL, the reward signal is used to evaluate how good an action is under a state, and we can transform hard-to-optimize objectives into maximizing the accumulative reward through reward design. Our objectives are twofold: maximizing the total fair throughput within limited energy and letting the UAV reach the destination before the energy is exhausted.

1) Fair Throughput Maximization: First, the reward at time slot \( n \) should include the fair throughput defined in (P2):

\[ r_{th}(n) = \kappa_{th}\hat{f}(n)R_c(n)\delta_t, \]

where \( \kappa_{th} \) is a positive constant that is used to adjust the reward of fair throughput maximization part.

2) Reach-Destination Task: For the reach-destination task, the straightforward rewards are very sparse, which means the UAV would be rewarded if and only if it reaches the destination when the battery runs out. Since the UAV’s battery could support its flying for thousands of time slots, the agent can receive just one signal indicating whether the UAV reaches the destination every thousands of time slots. In addition, the initial policy is randomly generated, and the UAV would arrive at the destination with probability almost zero. Accordingly, the agent could hardly learn the reach-destination task through straightforward rewards.

In order to deal with the sparse reward issue, we apply reward shaping [41] to make the rewards more trackable. The reward of reach-destination part is designed as

\[ r_{rd}(n) = \frac{d_{\text{dis}}}{[E(n)/\zeta_{rd}]r_{rd} + \epsilon_{rd}} \]

where \( d_{\text{dis}} \) denotes the reduced distance between the UAV’s current location and the destination after one-time-slot movement, \( \kappa_{rd} \) is the positive constant coefficient like \( \kappa_{th} \), \( \zeta_{rd} \) denotes the energy step size and \( \epsilon_{rd} \) is the value preventing the denominator from being zero. When the remaining energy \( E(n) \) is abundant, the reward of reach-destination part will be quite small and the agent focuses on maximizing the fair throughput. With more energy consumption, the agent focuses more on the reach-destination task.

Moreover, when the energy is exhausted, we should set a reward \( r_{ar} \) indicating whether the UAV arrives at the target spot, i.e.,

\[ r_{ar}(n) = \begin{cases} 0, & n = 1, \ldots, N_t - 1 \\ \xi_{ar}\kappa_{ar} + (1 - \xi_{ar})\kappa_{nar}, & n = N_t \end{cases} \]

where \( \xi_{ar} = 1 \) if the UAV arrives at the destination when the battery runs out, while \( \xi_{ar} = 0 \), otherwise. Furthermore, \( \kappa_{ar} \) is a positive constant to encourage arrival, while \( \kappa_{nar} \) is a negative constant to punish nonarrival.

3) Constraints: We set penalty rewards to punish the actions that violate the constraints (34) and (35):

\[ r_{ac}(n) = \xi_{ac,v}(n)\cdot\kappa_{ac}, \quad r_{al}(n) = \xi_{al,v}(n)\cdot\kappa_{al}, \]

where \( \xi_{ac,v}(n) \) is the binary acceleration constraint indicator with \( \xi_{ac,v}(n) = 1 \) implying the acceleration violates the constraint (34) and \( \xi_{ac,v}(n) = 0 \) otherwise. Similarly, \( \xi_{al,v}(n) \) is the binary altitude constraint indicator. The two negative constants \( \kappa_{ac} \) and \( \kappa_{al} \) are specific penalty rewards for the violation of constraints (34) and (35) respectively.

In summary, the shaped reward can be formulated as

\[ r(n) = r_{th}(n) + r_{rd}(n) + r_{ar}(n) + r_{ac}(n) + r_{al}(n). \]
E. Actor Network

The actor network takes the state $s(n)$ as input, and outputs the action $a(n)$. As shown in Fig. 4, we use the activation function $\text{sigmoid}$ to output $\lambda_v$, and $\lambda_{\varphi_p}$, $\tanh$ to output $\lambda_{\varphi_a}$, and $\text{softmax}$ to output $\lambda_b^k$, respectively, i.e.,

$$
\lambda_v = \text{sigmoid}(\chi_v),
$$
(50)

$$
\lambda_{\varphi_p} = \text{sigmoid}(\chi_{\varphi_p}),
$$
(51)

$$
\lambda_{\varphi_a} = \tanh(\chi_{\varphi_a}),
$$
(52)

$$
\lambda_b^k = (1 - \frac{KB_{\min}}{B})\text{softmax}(\{\chi_b^k\}) + \frac{B_{\min}}{B},
$$
(53)

where $\chi_v$, $\chi_{\varphi_p}$, $\chi_{\varphi_a}$, and $\{\chi_b^k\}$ are the pre-activation values of the corresponding neurons. There appear two specific problems: dimension imbalance and saturation. We propose dimension spread and pre-activation penalty to deal with these two problems respectively. Besides, we use a softmax reference technique to stabilize the training process.

1) Dimension Spread: According to Section III-B, most of the state dimensions are about location information, while only one dimension is about energy. Nevertheless, the energy dimension is very important since the UAV must reach the destination when the energy is exhausted. Thus there exists the dimension imbalance problem, and we spread the energy dimension to make it comparable to the location dimensions. As shown in Fig. 4, the energy dimension first connects to a spread network with size $1 \times N_e$ to extend the dimension to $N_e$, and then combines the other dimensions, e.g., uses’ locations, UAV’s location, UAV’s speed and UAV’s target location, to formulate the input of the actor network.

2) Pre-Activation Penalty: The activation functions sigmoid and tanh suffer from saturation problem. As shown in Fig. 5, when the absolute pre-activation value is greater than the tanh saturation value $\zeta_t$, i.e., in the saturation area, the output of activation function is approximately 1, and thus the back propagation process would face gradient vanishing problem. In order to avoid the saturation, we add the pre-activation penalty to the actor loss function, and (38) becomes

$$
L(\theta^\pi) = -Q(s(i), \pi(s; \theta^Q)) + \kappa_v \left(\max(\chi_v - \zeta_s, 0) + \max(-\chi_v - \zeta_s, 0)\right)^2
$$

$$
+ \kappa_{\varphi_p} \left(\max(\chi_{\varphi_p} - \zeta_s, 0) + \max(-\chi_{\varphi_p} - \zeta_s, 0)\right)^2
$$

$$
+ \kappa_{\varphi_a} \left(\max(\chi_{\varphi_a} - \zeta_t, 0) + \max(-\chi_{\varphi_a} - \zeta_t, 0)\right)^2,
$$
(54)

where $\kappa_v$, $\kappa_{\varphi_p}$ and $\kappa_{\varphi_a}$ are the coefficients of the pre-activation penalties for $\chi_v$, $\chi_{\varphi_p}$, and $\chi_{\varphi_a}$, and $\zeta_s$ is the sigmoid saturation value. The large pre-activation value of the neurons would cause large actor loss. In other words, minimizing the actor loss would let the pre-activation value stay in the unsaturation area.

3) Softmax Reference: The softmax activation function normalizes an input vector of $K$ real numbers into a probability distribution consisting of $K$ probabilities proportional to the exponentials of the input numbers. Note that it only makes
concatenates the neurons with fixed value outputs (stabilize the training process. Specifically, the actor network to a fixed value as the reference of the input in order to lead to the larger output. We set a pre-activation neuron and divergence since the larger pre-activation value may not is an all-zeros or all-ones vector, which may lead to oscillation For example, the output will be the same no matter the input.

{neurons, and we get the frequency band allocation action
sense when there are differences among these $K$ real numbers. For example, the output will be the same no matter the input is an all-zeros or all-ones vector, which may lead to oscillation and divergence since the larger pre-activation value may not lead to the larger output. We set a pre-activation neuron to a fixed value as the reference of the input in order to stabilize the training process. Specifically, the actor network outputs $(K - 1)$ neurons for the frequency band allocation and concatenates the neurons with fixed value 0 to the previous $(K - 1)$ neurons. Next, the softmax function activates the neurons, and we get the frequency band allocation action $\{\lambda_k^n\}_{k \in K}$.

**F. Critic Network**

As shown in Fig. 6, the critic network takes the state $s(n)$ and action $a(n)$ as input, and outputs the action-value $Q(s(n), a(n))$. The hidden layer architecture of critic network is the same as that of actor network. In addition, the dimension spread technique is retained in the critic network.

**G. Training Algorithm**

The EEFC-TDBA algorithm is episodic with each episode starting from the departure spot and ending up when the battery runs out.

In the training phase, the UAV’s departure and target locations as well as the GUs’ initial locations are initialized at the beginning of each episode. The UAV is stationary at first with fully-charged battery of energy $E_{max}$. The GUs move around on the ground, with their locations changing over time.

At each time slot, the UAV chooses action $a(n)$ through the actor network $\pi(s(n); \theta^\pi)$ and then adds an exploration noise $\mathcal{N}$ that is used to prevent the agent from falling into local optimal policy. We select the normal distribution noise with zero mean and deviation $\sigma_\mathcal{N}$. We also need to deal with violations of the altitude and acceleration constraints. If the action causes the violation of the altitude limit, then the altitude would be readjusted to the corresponding altitude limit boundary, and the agent receives a penalty reward, which has been expounded in Section III-D. Similarly, if the action causes the violation of acceleration constraint, then the UAV would accelerate or decelerate with acceleration magnitude $a_{max}$, which depends on the sign of the original acceleration. Then the agent obtains the next state $s(n + 1)$ and the reward $r(n)$, and stores the corresponding transition tuple $(s(n), a(n), s(n + 1), r(n))$ in the experience replay buffer $\mathcal{C}$. At each episode, we uniformly sample batches of experience from the replay buffer and update the networks through minimizing the actor and critic loss. Then the target networks are slowly updated according to Line 24 and Line 25 in Algorithm 1.

In the implementation phase, the UAV chooses the flight and frequency band allocation strategies through the well-trained actor network according to the current state.

**IV. Performance Evaluation**

In this section, simulations are conducted to evaluate the performance of the proposed EEFC-TDBA.

---

Fig. 6. The critic network architecture.

**Algorithm 1** EEFC-TDBA

1: Initialize the networks, including actor network $\pi(s; \theta^\pi)$, the target actor network with weights $\theta^\pi = \theta^\pi$, the critic network $Q(s, a; \theta^Q)$ and the target critic network with weights $\theta^Q = \theta^Q$.
2: Initialize the experience replay buffer $\mathcal{C}$
3: for each episode do
4: Initialize the locations of the UAV and the GUs, as well as the destination.
5: The UAV’s initial speed is zero, and the battery energy is $E_{max}$
6: for each time slot $n$ do
7: The UAV gets the GUs’ locations and formulate the state $s(n)$
8: $a(n) = \pi(s(n); \theta^\pi) + \mathcal{N}$, where $\mathcal{N}$ is exploration noise.
9: The UAV takes the action $a(n)$
10: if the action violates the altitude constraint then
11: Readjust the altitude to the corresponding altitude limit boundary
12: end if
13: if the action violates the acceleration constraint then
14: $v(n + 1) = v(n) + \xi_{ac}a_{max}$
15: end if
16: Update the state $s(n + 1)$ and obtain the reward $r(n)$
17: Store $\{s(n), a(n), r(n), s(n + 1)\}$ in the experience replay buffer $\mathcal{C}$
18: end for
19: Sample several random minibatches of $N_b$ transitions from $\mathcal{C}$
20: Calculate the critic target according to (37)
21: Update the critic network $\theta^Q$ by minimizing the critic loss (36)
22: Update the actor network $\theta^\pi$ by minimizing the actor loss (54)
23: Soft updates for the target networks:
24: $\theta^Q = \varepsilon \theta^Q + (1 - \varepsilon) \theta^Q$
25: $\theta^\pi = \varepsilon \theta^\pi + (1 - \varepsilon) \theta^\pi$
26: end for
uniform distribution on the interval $[\eta_{\text{LoS}}]$, while the speeds of the users have a specific reward setting. The altitude is set as $z = 300$ m and the lower limit is $z_{\text{min}} = 100$ m. The departure location of UAV is set as $u_0 = (-500, 0, 100)$ and the destination is $u_c = (500, 0, 100)$. The fully charged energy of the on-board battery is $E_{\text{max}} = 1 \times 10^5$ Joule. The maximum flying speed is $V_{\text{max}} = 20$ m/s, and the maximum acceleration magnitude is $a_{\text{max}} = 5$ m/s$^2$. There are $K = 10$ GUs sharing the total communication bandwidth $B = 1$ MHz with the noise power spectral density $n_0 = 10^{-17}$ W/Hz and the power for communication $P_e = 1$ W. The minimum bandwidth is set as $B_{\text{min}} = 5$ kHz. In each episode, the initial locations of the $K$ users are randomly generated in a $700 \times 700$ m square area with uniform distribution, while the speeds of the users have a uniform distribution on the interval [0 m/s, 5 m/s]. Each user follows a fixed mobility pattern that are taken from “straight line”, “circle”, and “triangle”. The initial directions of the users are also randomly distributed on the interval $[0, \pi]$. Since the state includes the GUs’ locations, the UAV can choose its action according to the GUs’ current location. As a result, the mobility pattern of GUs has no substantial impact on the performance. Furthermore, the channel-related parameters are $\eta_a = 12.08$, $\eta_b = 0.11$, $\eta_{\text{LoS}} = 1.6$ dB, and $\eta_{\text{NLoS}} = 23$ dB. The time slot is fixed as $\delta_t = 0.2$ s. The specific reward settings are shown in Table I.

### A. Simulation Settings

In our simulation, there is no limit on the horizontal coordinates of the UAV trajectory, while the upper limit of the altitude is set as $z_{\text{max}} = 300$ m and the lower limit is $z_{\text{min}} = 100$ m. The departure location of UAV is set as $u_0 = (-500, 0, 100)$ and the destination is $u_c = (500, 0, 100)$. The fully charged energy of the on-board battery is $E_{\text{max}} = 1 \times 10^5$ Joule. The maximum flying speed is $V_{\text{max}} = 20$ m/s, and the maximum acceleration magnitude is $a_{\text{max}} = 5$ m/s$^2$. There are $K = 10$ GUs sharing the total communication bandwidth $B = 1$ MHz with the noise power spectral density $n_0 = 10^{-17}$ W/Hz and the power for communication $P_e = 1$ W. The minimum bandwidth is set as $B_{\text{min}} = 5$ kHz. In each episode, the initial locations of the $K$ users are randomly generated in a $700 \times 700$ m square area with uniform distribution, while the speeds of the users have a uniform distribution on the interval [0 m/s, 5 m/s]. Each user follows a fixed mobility pattern that are taken from “straight line”, “circle”, and “triangle”. The initial directions of the users are also randomly distributed on the interval $[0, \pi]$. Since the state includes the GUs’ locations, the UAV can choose its action according to the GUs’ current location. As a result, the mobility pattern of GUs has no substantial impact on the performance. Furthermore, the channel-related parameters are $\eta_a = 12.08$, $\eta_b = 0.11$, $\eta_{\text{LoS}} = 1.6$ dB, and $\eta_{\text{NLoS}} = 23$ dB. The time slot is fixed as $\delta_t = 0.2$ s. The specific reward settings are shown in Table I.

### B. Network Architecture

The specific actor network architecture is shown in Fig. 4, where there are four hidden layers with 100, 150, 100, 50 neurons respectively. The spread dimension is set as $N_e = 10$, and thus the input layer has $3K + 7 + N_e = 47$ neurons. The network outputs the UAV speed, the UAV direction and the frequency band allocation strategy, and thus the output layer has $3 + K = 13$ neurons with one fixed reference neuron. The activation functions of all layers are all ReLU functions except for the output layer. Moreover, the critic network has the same hidden layers as actor network with input size $47 + 13 = 60$ and output size 1. The states of the UAV are normalized to $[0, 1]$. We employ ADAM optimizer [42] with a learning rate of $10^{-3}$ for both the actor and critic networks. The discount factor is $\gamma = 0.99$ and the soft update rate is $\epsilon = 0.005$. The exploration noise deviation is $\sigma_{\mathcal{N}} = 0.1$.

### C. Performance and Analysis

We compare EEFC-TDBA with two baseline methods, EEFC-TD and straight-flight.5

- EEFC-TD: This is a simplified version of EEFC-TDBA without frequency band allocation part, where we re-implement the trajectory design with the same reward settings and hyper-parameters as EEFC-TDBA.

- Straight-flight: The UAV flies straight to the destination. We calculate the speed so that the UAV can reach the destination exactly when it runs out of the energy. The frequency band allocation strategy for both EEFC-TD and straight-flight are randomly generated.

We compare EEFC-TDBA with the two baselines in terms of fair index, total throughput, fair throughput and minimum throughput. The minimum throughput represents the minimum throughput among GUs. We use the well-trained neural network to execute the energy-efficient fair communication service mission during the implementation phase and calculate these metrics for the entire episode. As illustrated in Table II, EEFC-TDBA outperforms both baselines for all metrics. Compared to straight-flight, EEFC-TD can improve the throughput via trajectory design from $2.151 \times 10^8$ to $4.235 \times 10^8$, i.e., a 96.9% improvement. Due to the 14.5% improvement of fair index, the increase in fair throughput is greater than that in throughput about 103.9%, and the minimum throughput

### TABLE I

<table>
<thead>
<tr>
<th>Reward Parameters</th>
<th>Simulation value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta_{\text{th}}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$\eta_{rd}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\epsilon_{rd}$</td>
<td>$7.5 \times 10^4$</td>
</tr>
<tr>
<td>$\epsilon_{rd}$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\gamma_{ar}$</td>
<td>$1.0 \times 10^4$</td>
</tr>
<tr>
<td>$\gamma_{mar}$</td>
<td>$-1.0 \times 10^3$</td>
</tr>
<tr>
<td>$\gamma_{sc}$</td>
<td>$-5.0$</td>
</tr>
<tr>
<td>$\gamma_{sd}$</td>
<td>$-5.0$</td>
</tr>
</tbody>
</table>

5To the best of the authors’ knowledge, there is existing paper considering the same scenario, and hence we cannot make a comparison with other works.
TABLE II
PERFORMANCE COMPARISON

<table>
<thead>
<tr>
<th>Metrics</th>
<th>EEFC-TDBA</th>
<th>EEFC-TD</th>
<th>Straight-flight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair index per episode</td>
<td>0.931</td>
<td>0.672</td>
<td>0.587</td>
</tr>
<tr>
<td>Total throughput per episode</td>
<td>$6.431 \times 10^8$</td>
<td>$4.235 \times 10^8$</td>
<td>$2.151 \times 10^8$</td>
</tr>
<tr>
<td>Fair throughput per episode</td>
<td>$5.308 \times 10^8$</td>
<td>$2.654 \times 10^8$</td>
<td>$1.308 \times 10^8$</td>
</tr>
<tr>
<td>Minimum throughput per episode</td>
<td>$3.695 \times 10^8$</td>
<td>$1.177 \times 10^8$</td>
<td>$3.705 \times 10^7$</td>
</tr>
</tbody>
</table>

Fig. 8. Effectiveness demonstrations of different techniques. No softmax reference means that we apply EEFC-TDBA without softmax reference technique. Similar for the others.

Fig. 9. Mean speed per episode with training.

Fig. 10. Probability of constraints violation.

Fig. 11. Fair Index vs number of GUs.

...can achieve an improvement of 217.7%. Combining the frequency band allocation to EEFC-TD, EEFC-TDBA can further improve the performance. For example, the fair index can be improved by 38.5%. Note that the closer the fair index is to 1, the harder it is to make further improvement. As a result, compared to trajectory design, frequency band allocation can significantly increase the fairness among the GUs. The total throughput for EEFC-TDBA is about 1.5 times that of EEFC-TD, while EEFC-TDBA has twice the fair throughput and more than three times the minimum throughput of EEFC-TD.

In Fig. 7, we plot the training curves for the accumulative reward, mean fair index, and total fair throughput of EEFC-TDBA and EEFC-TD. With frequency band allocation, EEFC-TDBA outperforms EEFC-TD with sufficient training. We observe an interesting phenomenon that in the first...
1000 episodes, the fair index and fair throughput of EEFC-TDBA are similar to that of EEFC-TD, but the accumulative reward of EEFC-TDBA is smaller than EEFC-TD and is even smaller than zero. This is because EEFC-TDBA needs to deal with the frequency band allocation problem in addition to trajectory design, which makes it more difficult to control the UAV flying to the destination. The UAV may fly far away from the destination, which leads to the negative reward of reach-destination part. The performance of EEFC-TDBA changes significantly when the number of the episode is around 1500. There are two reasons to explain this observation. First, DRL is different from supervised learning, since it has no clear label information. Thus the training curve of DRL often has obvious oscillation. Second, in an actor-critic based method, the training of actor network rely on the critic network, which means that a bad critic would lead to poor performance. When the training of critic network oscillates, the performance of EEFC-TDBA will change more significantly. After 2000 episodes of training, both EEFC-TDBA and EEFC-TD can control the UAV to reach the destination, and the accumulative reward starts to grow significantly. At 3000 episodes, the fair index of EEFC-TD has a remarkable growth, leading to an increase of the accumulative reward. On the other hand, EEFC-TDBA can further improve the performance, including fair index and fair throughput by frequency band allocation. After 4000 episodes of training, the accumulative reward of EEFC-TDBA converges without significant improvement.

In Fig. 8, we demonstrate the effectiveness of the techniques introduced in Section III-E. We can observe that EEFC-TDBA without pre-activation penalty has no improvement over the course of training. The pre-activation values of the neurons in the output layer are in the saturation area, and there is no effective gradient information for back propagation. Besides, without pre-activation penalty, UAV cannot learn to reach the destination when the battery is exhausted. Similarly, the UAV can not accomplish the reach-destination task without dimension spread technique due to the dimension imbalance problem. However, EEFC-TDBA without dimension spread can still improve the accumulative reward by increasing the reward of fair throughput part and achieve a 42.9% improvement compared with no pre-activation penalty. The lack of softmax reference slows the training down and results in a decrease of about 40% compared to EEFC-TDBA, but does not affect the learning of reach-destination task.

As illustrated in Fig. 9, the mean speed of EEFC-TD and EEFC-TDBA converge around 10 m/s to 16 m/s after 3000 episodes of training. As shown in Fig. 2, the propulsion power consumption is smaller in this speed range, which leads to longer flight time and larger accumulative reward.

As mentioned in II-C, the UAV trajectory should satisfy the acceleration and altitude constraints (34) and (35). Fig. 10 plots the violation probability of the acceleration and altitude constraints. We can see that at the beginning of training, the acceleration violation probability drops to almost zero, while the altitude violation probability decreases slowly with training. This is because the acceleration violation is only related to the speed $v$ in the output action, while whether the altitude constraint is violated depends on both the velocity $v$ and the UAV location $u$, which increases the difficulty of training. In addition, once the altitude constraint is violated, the UAV would be located at altitude limit boundary, which further increases the possibility of continuing to violate the altitude constraint. Note that the accumulative reward converges after 4000 episodes training, while the altitude violation probability converges to almost zero at 5000 episodes. The reason is that as the training goes on, the accumulative reward gets larger while the penalty reward of altitude violation gets
Fig. 13. Schematics of the main forces acting on the UAV.

smaller. In other words, the impact of the altitude violation on accumulative reward decreases significantly. After 4000 episodes of training, the accumulative reward reaches $1 \times 10^5$, while the altitude violation penalty reward is only about $1.5 \times 10^3$.

Fig. 11 illustrates the impact of the number of GUs on the fair index. We can observe that the fair index of two baselines decreases monotonously as the number of GUs increases, while EEFC-TDBA maintains the fair index at a high level. Increasing the number of GUs would cause more difficulty in fair communication service and it is hard to completely offset this growing difficulty only through trajectory design. As a result, the fair index of EEFC-TD decreases like straight-flight. Nevertheless, EEFC-TDBA can adjust frequency band allocation among GUs to provide fair communication service. Hence, the fair index of EEFC-TDBA will not decline significantly when the number of GUs increases.

The UAV trajectory is shown in Fig. 12, where both EEFC-TDBA and EEFC-TD can control the UAV to reach the destination when the energy is exhausted. Note that the UAVs will raise their altitudes first. This is because the higher altitude leads to greater probability of obtaining LoS channel and smaller pathloss. It can be seen that the trajectories of both EEFC-TDBA and EEFC-TD are irregular. This is because these two methods choose flight direction and speed according to the locations of GUs. Since GUs are randomly distributed on the ground and move around, the trajectory of a better design should intuitively “match” the distribution of GUs and would look more irregular. We can also see that the trajectories of EEFC-TDBA are smoother than that of EEFC-TD. This is due to the fact that the output action of EEFC-TDBA also includes the frequency band allocation, which affects the stability of the output of trajectory part.

V. CONCLUSION

In this paper, we have proposed a DRL based algorithm EEFC-TDBA for 3D UAV trajectory design and frequency band allocation, which enables the UAV to provide energy-efficient and fair communication service. Specifically, the UAV chooses its flight speed, direction and frequency band allocation strategy based on the current location, the current speed, the destination, the remaining energy of UAV, and the GUs’ locations. EEFC-TDBA maximizes the fair throughput within limited on-board energy to balance between the throughput maximization and the fairness among the GUs. Simulation results have demonstrated that EEFC-TDBA can achieve much better performance than EEFC-TD and straight-flight in terms of the total throughput and the fairness among the GUs. For the future work, we will investigate the trajectory design and resource allocation of multiple UAVs to provide better communication service for the GUs.

APPENDIX A

ENERGY CONSUMPTION MODEL FOR QUAD-ROTOR UAV IN 3D ENVIRONMENT

In the appendix, we formulate the energy consumption model for a quad-rotor UAV in 3D flight.6 We model the

<table>
<thead>
<tr>
<th>Notation</th>
<th>Physical meaning</th>
<th>Simulation value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>UAV mass in kg</td>
<td>2</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity acceleration in m/s$^2$</td>
<td>9.8</td>
</tr>
<tr>
<td>$l_e$</td>
<td>Rotor radius in meter (m)</td>
<td>0.1</td>
</tr>
<tr>
<td>$A$</td>
<td>Disc area for each rotor in m$^2$</td>
<td>0.0314</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Blade angular velocity in radian/second</td>
<td>—</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Air density in kg/m$^3$</td>
<td>1.293</td>
</tr>
<tr>
<td>$c_T$</td>
<td>Thrust coefficient based on disc area, $T_h = c_T \rho A \Omega^2 l_e^2$</td>
<td>0.302</td>
</tr>
<tr>
<td>$c_w$</td>
<td>Weight coefficient for each rotor, $c_w = \frac{mg}{n_r \rho c_e A \Omega^2 l_e^2}$</td>
<td>—</td>
</tr>
<tr>
<td>$n_r$</td>
<td>Number of rotors</td>
<td>4</td>
</tr>
<tr>
<td>$n_b$</td>
<td>Number of blades</td>
<td>2</td>
</tr>
<tr>
<td>$l_c$</td>
<td>Blade chord length in meter (m)</td>
<td>0.015</td>
</tr>
<tr>
<td>$c_s$</td>
<td>Rotor solidity, $c_s = \frac{n_b l_c}{2 \pi l_e}$</td>
<td>0.0955</td>
</tr>
<tr>
<td>$S_{FP}$</td>
<td>Fuselage equivalent flat plate area in m$^2$</td>
<td>0.01</td>
</tr>
<tr>
<td>$d_0$</td>
<td>Fuselage drag drag ratio for each rotor, $d_0 = \frac{S_{FP}}{A}$</td>
<td>0.834</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Local blade section drag coefficient</td>
<td>0.012</td>
</tr>
<tr>
<td>$c_f$</td>
<td>Incremental correction factor to induced power relative to that for linear induced velocity</td>
<td>0.131</td>
</tr>
<tr>
<td>$v$</td>
<td>Forward velocity of the quad-rotor UAV in m/s</td>
<td>—</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Forward speed normalized on tip speed, $\delta = \frac{v}{\frac{1}{2} \Omega l_e}$</td>
<td>—</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Advance ratio, $\mu \approx \delta$</td>
<td>—</td>
</tr>
<tr>
<td>$v_0$</td>
<td>Thrust velocity, $v_0 = \sqrt{\frac{2 \rho T_h}{\rho c_e A \Omega^2 l_e^2}}$</td>
<td>—</td>
</tr>
<tr>
<td>$\nu_0$</td>
<td>Mean induced velocity</td>
<td>—</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>Mean induced velocity normalized on tip speed, $\lambda_i = \frac{\mu v_0}{\frac{1}{2} \Omega l_e}$</td>
<td>—</td>
</tr>
<tr>
<td>$D$</td>
<td>Drag of fuselage, which is in the opposite direction of the UAV velocity, $D = \frac{1}{2} \rho u^2 S_{FP}$ Equation (4.5) in [27]</td>
<td>—</td>
</tr>
<tr>
<td>$\tau_c$</td>
<td>Climb angle</td>
<td>—</td>
</tr>
<tr>
<td>$T_h$</td>
<td>Thrust of each rotor</td>
<td>—</td>
</tr>
<tr>
<td>$t_{c,D}$</td>
<td>Thrust coefficient referred to disc axes, $t_{c,D} = \frac{T_h}{\rho c_e A \Omega^2 l_e^2}$</td>
<td>—</td>
</tr>
<tr>
<td>$q_c$</td>
<td>Torque coefficient, which is related to the power for each rotor. The total power for the UAV is $P = \sum_{i=1}^{n_r} q_c \rho c_e A \Omega^2 l_e^2$</td>
<td>—</td>
</tr>
</tbody>
</table>

6Inspired by the energy consumption model for single-rotor UAV in 2D flight in [27], we derive a energy consumption model for quad-rotor UAV in 3D environment.
The propulsion power for a quad-rotor UAV can be calculated by

$$P(v, T_h) = n_r q_c \rho c_s A \Omega^3 l_r^3$$

where $n_r$, $\rho$, $c_s$, and $A$ are constants. The torque coefficient $q_c$ is given by [43, (4.20)]

$$q_c = \frac{\delta}{8} (1 + 3\mu^2) + (1 + cf) \lambda_{l_c,D} + c_w \hat{v} \sin \tau_c + \frac{1}{2} d_0 \hat{v}^3.$$  

(56)

By substituting $\mu \approx \hat{v} = \frac{v}{\Omega r}$, $l_c,D \approx \frac{T_h}{\rho c_s A^2 \Omega r}$, $\lambda_i$, we have

$$v_{th} = \frac{m g}{n_r \rho c_s A \Omega^3 l_r^3}$$

into (56), $q_c$ can be rewritten as

$$q_c = \frac{\delta}{8} (1 + 3\mu^2) + (1 + cf) \left(\frac{\sqrt{v_0^4 + \mu^4 - \mu^2}}{\sqrt{\frac{v_0^4}{\mu^2} - \frac{\mu^4}{2}}}\right) \frac{1}{2} T_h$$

+ $m g v \left(\frac{\rho c_s A^2 l_r^3}{n_r \rho c_s A \Omega^3 l_r^3}\right) \sin \tau_c + \frac{1}{2} d_0 \hat{v}^3.$$

(57)

Then the propulsion power can be expressed as a function of UAV speed $v$ and thrust for each rotor $T_h$, shown at the top of the page. According to $T_h = c \rho A^2 \Omega^2 l_r^2$, the propulsion power can be re-written as (11).

We use force analysis on the UAV to establish the relationship between the thrust and the UAV movement. As shown in Fig. 13, the UAV is affected by gravity, air drag, and the thrust from the rotors. According to Newton’s second law, we have

$$m a_c = n_r T_h + D + m g$$

(59)

where $|D| = \frac{1}{2} \rho v^2 S_{FP}$ is the air drag and is in the opposite direction of the UAV’s velocity. Therefore, we obtain (10) as shown in Section II-B.


