Intelligent Trajectory Design in UAV-aided Communications with Reinforcement Learning

Sixing Yin, Shuo Zhao, Yifei Zhao, and F. Richard Yu

***Abstract*—In this paper, we focus on a cellular network aided an unmanned aerial vehicle (UAV) that serves as an aerial basestation for multiple ground users. The UAV's trajectory design is investigated to maximize the expected uplink sum rate with inaccessibility to user-side information, such as locations and transmit power, and channel parameters. The problem is formulated as Markov decision process (MDP) and solved with model-free reinforcement learning. Due to the continuous and deterministic action space, the deterministic policy gradient (DPG) algorithm is applied for the reinforcement learning model. Experiment results show that due to the great generalizability of the reinforcement learning model, the UAV is able to intelligently track the ground users with the learned trajectory despite being unaware of the user-side information and channel parameters, even when the ground users are mobile. The performance of the learned trajectory is fairly close to that of the optimal trajectory derived through conventional optimization problem solving with such information explicitly known.**

***Index Terms*—UAV-assisted communications, trajectory design, reinforcement learning.**

I. Introduction

Due to the remarkable features of cost effectiveness, high mobility and deployment flexibility, unmanned aerial vehicles (UAVs) have been increasingly popular in recent years for a variety of specialized applications. In particular, thanks to their better capability of finding line-of-sight (LoS) spots, UAVs equipped with radio transceivers have been considered as one of the effective options for the next-generation wireless communications, which shift network deployment from ground plane to aerial space [1].

Compared to traditional wireless communications with terrestrial infrastructures, one salient feature of UAV-aided communications is that UAVs are able to flexibly adjust their locations to maintain favorable channel condition. In this sense, UAVs' trajectory design is non-negligible for system performance enhancement [2]. There have been volumes of related work that involve trajectory design in a variety of scenarios, including throughput maximization in mobile relaying [3], operation time minimization in cellular networks [4], energy saving in data transferring [5] as well as energy transfer fairness in UAV-enabled wireless power transfer [6].

In most of the existing work on trajectory design for UAV-aided communications, it is assumed that the locations of ground receivers are perfectly known such that the channel gain can be explicitly derived by following a specific radio propagation model and trajectory design can be conducted through conventional optimization problem. However, in practice, exact system information might not be easily accessible since dedicated communication resources are not always be available, especially for user locations and channel parameters, which could be unpredictable due to their dynamics.

Model-free reinforcement learning framework has been an efficient solution to tackle such an issue, where the best decision is "learned" only by interacting with environment in a "try-and-error" fashion without an explicit model, and its application can be extensively found in numerous areas such as robotics and economics. Recently, there has also been work on reinforcement learning applied to sequential decision making problems in wireless communication systems [7]–[9].

In this paper, we focus on uplink transmission of a cellular network with frequency division multiple access, where an UAV serves as an aerial basestation for multiple ground users. Different from most of the existing work on trajectory design for UAV-aided communications, the UAV is inaccessible to user-side information, i.e., locations and transmit power, and channel parameters. The UAV's trajectory is designed to optimize the expected uplink sum rate by leveraging reinforcement learning such that it is able to intelligently track the ground users. Deterministic policy gradient (DPG) algorithm is applied since the desired policy and action are supposed to be deterministic and continuous. We show that with the proposed trajectory, the UAV is able to track the ground users even if they are mobile and the performance is close to that via conventional optimization problem solving, where user-side information and channel parameters are perfectly known.

II. Problem Statement

We focus on a uplink cellular network, where a UAV travelling in the air at altitude H serves as an aerial basestation for K ground users through frequency division multiplexing, i.e., each ground user transmits on its own assigned channel. Inter-channel interference is not considered since it can be largely eliminated by existing techniques (i.e., guard band). Without a dedicate channel for signalling, the UAV is unable to access the exact information on location and transmit power of the ground users as well as channel condition parameters (e.g., path loss exponent) and can only observe the received signal transmitted by each ground user.

In order to optimize the overall uplink transmission performance, the UAV's trajectory along which it travels to the ground users has to be properly designed. In order for ease of derivation, we consider N steps for the entire trajectory by discretizing the entire time horizon into T equal-spacing timeslots and assume that the UAV's location within each timeslot, denoted by $\boldsymbol{q} = \{\boldsymbol{q}_1, ..., \boldsymbol{q}_T\}$, is approximately constant. Thus the problem of uplink sum rate optimization can be formulated as

$$\begin{aligned} \max_{\boldsymbol{q}} \quad & J=\sum_{t=1}^{T}\sum_{k=1}^{K}\log(1+\frac{p_k\gamma_0}{\sigma||\boldsymbol{q}_t-\boldsymbol{w}_k||^{\beta}}) \\ \text{s.t.} \quad & ||\boldsymbol{q}_t-\boldsymbol{q}_{t-1}||\leq V, \forall t=1,..,T \end{aligned} \tag{1}$$

where p_k and $\boldsymbol{w}_k$ are the uplink transmit power and location of user k, σ is the channel noise power, γ_0 is the reference channel gain, β is the path loss exponent and V is the UAV's maximum travel distance within each timeslot. Specially, $\boldsymbol{q}_0$ and $\boldsymbol{q}_T$ refer to the UAV's start and end locations, respectively. The problem in (1) can be readily solved by following the successive optimization technique in [3]. However, due to unawareness of user-side information as well as channel condition parameters, solving such a problem through conventional optimization techniques is almost impossible since it is unable to be explicitly formulated. In this case, the UAV should manage to intelligently steer its way toward the ground users according to its observations.

Since the UAV is unable to predict the exact uplink rate for all the ground users, one intuitive solution for its trajectory design is learning in a "try-and-error" way. Specificly, the UAV may first try a few steps of movement, then make evaluation with the feedback (e.g., whether the received signal turns stronger or weaker) to see whether the movement is a wise action. Such a learning process falls into a typical reinforcement learning framework.

III. Reinforcement Learning Solution

Obviously, the UAV's trajectory can be designed as a sequential decision making process, i.e., movement at a single step is determined each time based on the current situation. In this sense, Markov decision process (MDP), which aims at finding the best policy (a mapping function from the current situation to the best decision), suits this problem well.

A. MDP formulation

In general, an MDP $\mathcal{M}$ can be defined by five elements, $\mathcal{M}=\{\mathcal{S},\mathcal{A},P,R,\gamma\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the state transition probability, R is the reward (also known as cost) at each step and γ is the discount factor. Here we have $\gamma=1$ since no discount is imposed on reward in future according to (1).

1) State: In an MDP, the system state should be observable and accessible, which makes the UAV's received signal strength eligible. Instead of the instantaneous received signal strength in each timeslot, we resort to its temporal difference to define the system state because temporal difference is able to indicate changes in distance from the UAV to each ground user and whether an action is wise enough. Let $\boldsymbol{e}_t=\{e_{1,t},...,e_{K,t}\}$ denote the UAV's received signal strength on all the channels (from all the ground users) in timeslot t and $\boldsymbol{\Delta e}_t=\boldsymbol{e}_t-\boldsymbol{e}_{t-1}$ denote the temporal difference. Then we define system state as the temporal difference in L timeslots, i.e., $\boldsymbol{S}_t=\{\boldsymbol{\Delta e}_t,\boldsymbol{\Delta e}_{t-1},...,\boldsymbol{\Delta e}_{t-L+1}\}$. Here more than one timeslots are included for definition of the system state to make a better decision on movement at each step by utilizing the state transition history.

2) Action: The UAV's action can be defined as its movement in each timeslot. With polar coordinates, we define $\boldsymbol{A}_t=\{z_t,\theta_t\}$ as the action taken in timeslot t, i.e., the UAV's movement with polar coordinates representation, where $z_t\in[0,V]$ is the step size and $\theta_t\in[0,2\pi]$ is the direction angle. With such a definition, the constraint for UAV's maximum travel distance within each timeslot is always satisfied.

3) Reward: Since we focus on an MDP with finite time horizon, i.e., T steps of movement, the reward at each step can be well defined by the uplink sum rate in each timeslot, which is given by $R_t=\sum\limits_{k=1}^{K}\log[1+p_k\gamma_0/(\sigma||\boldsymbol{q}_t-\boldsymbol{w}_k||^{\beta})]$. However, the UAV observes only R_t (feedback from the environment) while exact information on p_k, γ_0, σ, β and $\boldsymbol{w}_k$ is inaccessible. In addition, the state-action value function $Q^{\pi}(\boldsymbol{S}_t,\boldsymbol{A}_t)$ can be defined as the expected uplink sum rate with policy π at state $\boldsymbol{S}_t$ when action $\boldsymbol{A}_t$ is taken.

4) State Transition Probability: The state transition probability, denoted by $P(\boldsymbol{S}_{t+1}|\boldsymbol{S}_t,\boldsymbol{A}_t)$, is defined as the probability distribution of the next state given the current state and action taken and characterizes dynamics of the whole system. However, due to the continuous state space and action space, the exact state transition probability can never be obtained even by long-term statistics such that the optimal policy cannot be derived via off-line techniques, e.g., value iteration. Therefore, we resort to reinforcement learning, through which the optimal policy that maximizes the expected uplink sum rate $\mathbb{E}[\sum\limits_{t=1}^{T}R_t]$ is learned by interacting with the environment in a try-and-error fashion without exact state transition probability.

B. Deterministic Policy Gradient

Apparently, the action space of the MDP defined in Section III-A is continuous and the policy must be deterministic rather than stochastic as a probability distribution. Therefore, we resort to the DPG algorithm to solve the MDP, which falls into the actor-critic framework that comprises an interactive pair of policy and value networks to improve the learning efficiency of pure policy-based algorithms [10].

Experience replay and target networks are employed as well to decorrelate different training episodes and increase learning stability [11]. Moreover, a feature network is introduced preceding the policy network to extract the hidden temporal features since more than one timeslots are considered for the system state. The DPG algorithm for the UAV's trajectory design is summarized in Algorithm 1, where F denotes the feature network, Q and Q' denote main and target value networks, μ and μ' denote main and target value networks, and $\boldsymbol{\theta}$ denotes their parameters. The whole DPG algorithm consists of two stages: experience replay and model training.

In the first stage, each experience episodes (training data) are generated by taking actions (T steps for each episode) computed by the policy network with randomly initialized parameters and tuple $\{\boldsymbol{S}_t,\boldsymbol{A}_t,R_t,\boldsymbol{S}_{t+1}\}$ is stored in the replay buffer at each step. Here actions computed by the policy network are scrambled by an additive normal-distributed noise to increase experience diversity. The flowchart of experience replay is depicted in Fig. 1.

Algorithm 1 DPG for Trajectory Design

```
Randomly initialize the parameters of main value and policy networks as well as the feature network;
Initialize the parameters of target value and policy networks as $\boldsymbol{\theta}_{Q'} = \boldsymbol{\theta}_Q$ and $\boldsymbol{\theta}_{\mu'} = \boldsymbol{\theta}_\mu$;
for $m = 1, ..., M$ do
    Randomly generate ground user locations;
    Randomly choose action for $t = \{1, ..., L\}$
    for $t = L + 1, ..., T$ do
        Choose and execute action $\boldsymbol{A}_t = \mu(F(\boldsymbol{S}_t)) + N_t$ and observe a new state $\boldsymbol{S}_{t+1}$ and reward $R_t$;
        Store tuple $\{\boldsymbol{S}_t, \boldsymbol{A}_t, R_t, \boldsymbol{S}_{t+1}\}$ in the replay buffer;
        if The replay buffer size is sufficiently large then
            Sample a minibatch from the replay buffer;
            Update the parameters of feature network and main value network by minimizing (2);
            Update the parameters of feature network and main policy network based on (3);
            if $m$ is a multiple of $C$ then
                Update the parameters of target value and policy networks as $\boldsymbol{\theta}_{Q'} = \boldsymbol{\theta}_Q$ and $\boldsymbol{\theta}_{\mu'} = \boldsymbol{\theta}_\mu$;
            end if
        end if
    end for
end for
```

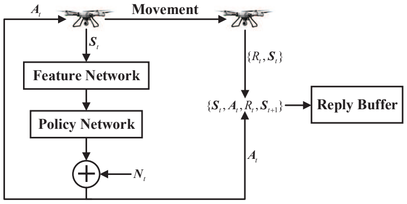

Fig. 1. Flow chart of experience replay in the DPG algorithm.

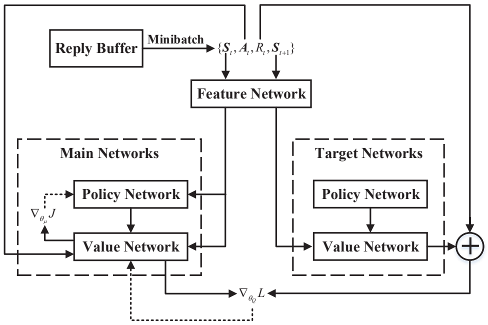

Fig. 2. Flow chart of model training in the DPG algorithm.

In the second stage, once sufficient experience episodes have been collected in the replay buffer, the main policy and value networks are alternately trained with minibatches randomly sampled from the replay buffer in each iteration while the parameters of the target networks are updated as a duplicate of the main networks only every C iterations. For the main value network, the parameters are updated in gradient-based manner to minimize the time difference error given by

$$L = \frac{1}{2N} \sum_{i=1}^{N} [y_i - Q(F(\boldsymbol{S}_i), \boldsymbol{A}_i)]^2 \tag{2}$$

as the loss function, where N is the batch size and $y_i = R_i + \gamma Q'[F(\boldsymbol{S}_{i+1}), \mu'(F(\boldsymbol{S}_{i+1}))]$. For the main policy network, the parameters are updated following the policy gradient in [10]:

$$\nabla_{\boldsymbol{\theta}_\mu} J = \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}_\mu} \mu(F(\boldsymbol{S}_i)) \nabla_{\boldsymbol{A}_i} Q(F(\boldsymbol{S}_i), \boldsymbol{A}_i) \tag{3}$$

to maximize the expected uplink sum rate. The flowchart of model training is depicted in Fig. 2, where the dashed arrows refer to gradient-based update for the parameters of value and policy networks. Specificly, the parameters of the main value and policy networks are alternately updated in each iteration with the gradient-based method (descent for the value network while ascent for the policy network). The parameters of the feature network are always updated in a similar way along with parameters of either main value network or policy network.

IV. Simulation Results

We focus on a square area of 100m-by-100m, where 15 ground users are located. The UAV travels at altitude 30m and always starts from $\boldsymbol{q}_0 = \{10, 10\}$. In each episode, ground users are randomly distributed within the square area of $\{[80, 100], [80, 100]\}$ and the training data is generated by the UAV taking T steps of actions, where we have $T = 75$ and $L = 3$ throughout the experiments. The optimal trajectory derived, which can be readily derived by solving (1) with the successive optimization technique in [3], is involved as a baseline for performance comparison with all information on location and transmit power of the ground users as well as channel condition parameters explicitly known. For system parameters, we have $p_k = 1$ for $k \in \{1, ..., K\}$, $V = 2$m, $\alpha = 2$ and $\gamma_0/\sigma = 1$. For reinforcement learning settings, both the value and policy networks are built with double-layered neural networks[1] and the feature network is built with recurrent neural network. With the replay buffer size 15000, the batch size 64 and the learning rate 10^{-6}, the reinforcement learning model is trained for 10000 episodes[2].

The performance of the learned trajectory is first evaluated with ground users staying constantly within a square area. We first have a basic test with homogeneous user distribution, i.e., the ground users are still distributed within the square area of $\{[80, 100], [80, 100]\}$, as experience episodes generation. Comparison between the learned trajectory and the optimal one is shown in Fig. 3. It can be seen that with the optimal trajectory, the UAV flies to a proper spot at the maximum speed straight within the square area, where the uplink sum rate can be maximized, and then hovering right above that spot. In contrast, with the learned trajectory, the UAV takes a detour but is still able to reach a spot within the ground user area in

[1] Sigmoid is used as the activation function for the output of the policy network, which results in a two-dimensional vector ranging from 0 to 1. To accord with the UAV's action space, the output is simply scaled with factor V for step size and 2π for direction angle to generate the actual action.

[2] Since we consider the temporal difference of received signal strength in L timeslots as the system state, the UAV's first L steps of actions are always randomly generated.

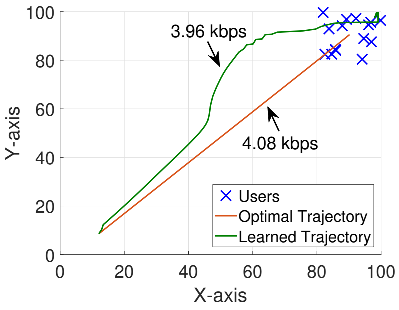

Fig. 3. Performance comparison for the learned and optimal trajectories with homogeneous user distribution.

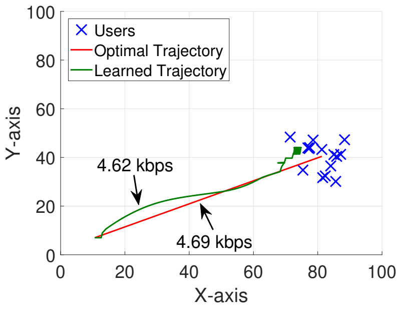

Fig. 4. Performance comparison for the learned and optimal trajectories with heterogeneous user distribution.

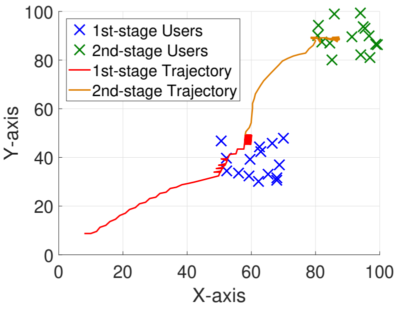

Fig. 5. Two-stage mobile user tracking with the learned trajectory.

spite of lack of information on location and transmit power of the ground users as well as channel condition parameters.

To validate the generalizability of the reinforcement learning model, we also test the UAV's learned trajectory with heterogeneous user distribution in a different square area of $\{[70, 90], [30, 50]\}$, as shown in Fig. 4. Similarly, with the learned trajectory, the UAV is still able to efficiently "track" the ground users with just a slight detour, which indicates great generalizability. The reason for this is that temporal difference of the UAV's received signal strength in consecutive timeslots implicitly indicates how good is the UAV's action at each step and also guides the UAV to adjust its actions accordingly. This is also why the UAV is able to travel back to the ground user from a detour, as shown in both Figs. 3 and 4.

From Figs. 3 and 4, we can see that although the UAV travels on a detour, the performance of the learned trajectory is not significantly outperformed by the optimal one, which also shows the high efficiency of the reinforcement learning model trained with DPG algorithm with inaccessibility to user-side information. Another finding is that, as the UAV flies into the ground user area, it intelligently reduces its step size while turns less decisive with a "zig-zag" trajectory. This is because when the UAV is located above the ground user area, different direction angles have minor effect on the uplink sum rate. In this case, the policy network makes state and direction angle to less correlated with each other, which results in an indecisive trajectory. This is especially the case where the ground users are located far away from the UAV's start location.

We further evaluate the tracking ability of the reinforcement learning model with mobile ground users. A two-stage test is performed, in which the 15 ground users are located within the area of $\{[70, 90], [30, 50]\}$ in the first stage, then move to the area of $\{[80, 100], [80, 100]\}$ in the second. The UAV's corresponding two-stage trajectory is shown in Fig. 5. We can see that despite being unaware of the exact locations of the ground users, the UAV is still be able to intelligently track them as they move due to the great generalizability of the reinforcement learning model with the DPG algorithm.

V. Conclusion

In this paper, trajectory design in a UAV-aided cellular network is investigated to maximize the the expected uplink sum rate without exact information on location and transmit power of the ground users as well as channel condition parameters. With MDP formulation, we resort to reinforcement learning model with deterministic policy as solution due to the continuous and deterministic action space. It is shown that, thanks to the great generalizability of the reinforcement model, the UAV is able to intelligently track the ground users with the learned trajectory even when the ground users are mobile. Moreover, the learned trajectory is not much worse in performance than that of the optimal trajectory with the exact information on location and transmit power of the ground users as well as channel condition parameters perfectly known.

References

[1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.

[2] Q. Wu, L. Liu, and R. Zhang, "Fundamental tradeoffs in communication and trajectory design for uav-enabled wireless network," 2018. [Online]. Available: http://arxiv.org/abs/1805.07038

[3] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for uav-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, Dec 2016.

[4] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in uav-enabled multicasting," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2233–2246, April 2018.

[5] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," 2018. [Online]. Available: http://arxiv.org/abs/1804.02238

[6] J. Xu, Y. Zeng, and R. Zhang, "Uav-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5092–5106, Aug 2018.

[7] M. Simsek, M. Bennis, and . Gven, "Learning based frequency- and time-domain inter-cell interference coordination in hetnets," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4589–4602, Oct 2015.

[8] A. Asheralieva, "Bayesian reinforcement learning-based coalition formation for distributed resource sharing by device-to-device users in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5016–5032, Aug 2017.

[9] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, Jan 2018.

[10] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14, 2014, pp. I–387–I–395.

[11] 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.