Meta Learning for Meta-Surface: A Fast Beamforming Method for RIS-Assisted Communications Adapting to Dynamic Environments

Qinpei Luo, Boya Di
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
{luoqinpei, diboya}@pku.edu.cn

***Abstract*—Recently reconfigurable intelligent surface (RIS) has been proposed as a promising technique to enhance the capacity of wireless networks by reshaping the electromagnetic characteristics of the environment. However, given numerous RIS elements, it is non-trivial to design an efficient beamforming scheme especially for the real-time mobile applications that require fast response to varying environments. In this paper, aiming to maximize the sum rate of a multi-user system via the RIS-enabled beamforming design, a meta-critic network is proposed to recognize the environment change and automatically perform the self-updating of the learning model. We also develop a stochastic Explore and Reload procedure to alleviate the high-dimensional action space issue. Simulation results demonstrate that the proposed scheme can converge to a higher sum rate more rapidly compared to the state-of-the-art methods in dynamic settings. The robustness of our proposed scheme against different RIS sizes is also verified.**

I. Introduction

As a novel enabling technology for the next-generation communications, meta surfaces is able to improve spectrum efficiency, among which reconfigurable intelligent surface (RIS) has attracted great attention owing to its ability of desirable signal reflection. Users can be well-served through programming the phase shifts of the RIS elements. To deal with the large number of RIS elements, machine learning based beamforming schemes have been explored, among which reinforcement learning (RL) has served as a potential tool to depict the interaction process between the surface and the environment. However, traditional RL methods may be time-consuming due to its re-training process, making it not feasible especially for the real-time dynamic environments.

In this paper, we propose a meta-critic deep deterministic policy gradient (MC-DDPG) scheme for the RIS-based beamforming adapting to dynamic user's locations. A meta-critic is designed which serves as an automotive tool for real-time model parameter generation in new environments by learning from multiple scenario-specific tasks. Simulation results show that **with only a small amount of cascaded channel information, MC-DDPG outperforms the traditional RL method and an iterative algorithm in terms of the sum rate and the convergence speed in dynamic environments. The robustness of the MC-DDPG scheme against different RIS sizes is also verified, which reveals the feasibility of large-scale MIMO enabled by RIS.**

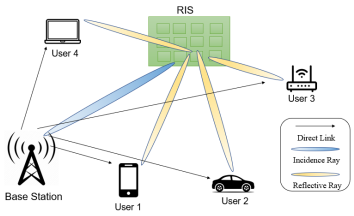

Fig. 1. System model of the RIS-assisted multi-user system

II. System Model

Fig. 1 shows a downlink multi-user MISO wireless communication system with a M-antenna base station (BS) serving K users each equipped with one antenna. The Light-of-Sight (LoS) channel between the BS and users are often unstable and suffering from severe fading. An RIS consisting of N elements is deployed between the users and the BS to reflect the transmit signals towards the users.

The direct channel between the BS and K users can be denoted as $\mathbf{H}_{BU} \in \mathbb{C}^{K\times M}$. The BS-RIS link and the RIS-user link can be denoted by $\mathbf{H}_{BI} \in \mathbb{C}^{N\times M}$ and $\mathbf{H}_{IU} \in \mathbb{C}^{K\times N}$, respectively. We assume that each channel of $\mathbf{H}_{BU}, \mathbf{H}_{BI}, \mathbf{H}_{U}$ follows the Markov process with the transition of time slots. The equivalent end-to-end channel from the BS to user k can be given as

$$\mathbf{H}_k = \mathbf{H}_{IU,k}\mathbf{\Theta}\mathbf{H}_{BI} + \mathbf{H}_{BU,k}, k \in \mathcal{K}_u, \tag{1}$$

in which $\mathbf{\Theta} \in \mathbb{C}^{N\times N} = diag([e^{j\theta_1}, \cdots, e^{j\theta_N}])$, $[e^{j\theta_1}, \cdots, e^{j\theta_N}]$ being the phase shifts configuration of RIS elements.

III. Problem Formulation And Algorithm Design

A. Sum Rate Maximization Problem

We consider the sum rate maximization problem in T time slots, each of which has a duration of ΔT. The signal-to-noise (SNR) and data rate of user k can be expressed as

$$\gamma_{k,t} = \frac{|(\mathbf{H}_{IU,k}\mathbf{\Theta}\mathbf{H}_{BI} + \mathbf{H}_{BU,k})\boldsymbol{V}_{k,t}m_k|^2}{|(\mathbf{H}_{IU,k}\mathbf{\Theta}\mathbf{H}_{BI} + \mathbf{H}_{BU,k})\sum_{j=1,\neq k}^{K}\boldsymbol{V}_{j,t}m_j|^2 + \sigma_{k,t}^2}, \tag{2}$$

$$R_{k,t} = |\Delta T log(1+\gamma_{k,t})|. \tag{3}$$

$\boldsymbol{V}_{j,t}$ refers to the digital beamforming vector from the BS to the j-th user, m_k denotes the symbol BS sends to user j, and $n_{k,t}$ represents Gaussian noise which follows $N(0, \sigma_{k,t}^2)$.

For a fixed digital beamforming scheme such as ZF [1] or MMSE, we can get a sub-optimal solution of $\boldsymbol{V}_t$ directly, the sum rate maximization problem can be formulated as

$$\max_{\boldsymbol{\Theta}_t} \sum_{t=1}^{T} \sum_{k=1}^{K} R_{k,t}, \tag{4}$$

Given the time-varying characteristics of channels, we then reformulate it as a MDP consisting of the following components: 1)**Action**:$a_t = \boldsymbol{\Theta}_t, \forall \theta_t \in \boldsymbol{\Theta}_t, \theta_t \in (-\pi, \pi)$.2) **State**:$s_t = \{\mathbf{H}_t, \boldsymbol{\Theta}_{t-1}\}$,where $\mathbf{H} = \mathbf{H}_{IU}\boldsymbol{\Theta}\mathbf{H}_{BI} + \mathbf{H}_{BU}$. 3) **Reward**:$r_t = \eta \sum_{k=1}^{K} \gamma_{k,t}$, where η is a coefficient.

B. MC-DDPG Algorithm Design

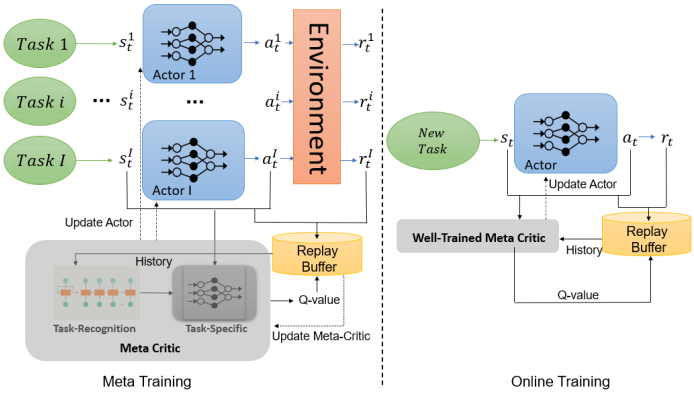

Fig. 2. Framework of MC-DDPG

The design of our proposed MC-DDPG is shown in Fig. 2, consisting of the meta learning phase and online learning phase. The experience from all actors will be sent to one meta critic for updating. The task in it refers to a process of the BS maximizing the sum rates of all users. For different tasks, the channel states and locations of users are different.

The meta learning phase can be described as follows: First, for each task i, its current state s_t^i is fed to actor i so the actor can generate an action a_t^i from policy $\pi(a|s_t^i)$. Task i operates the action and receives the reward r_t^i from the environment, then the transition tuples $< s_t^i, a_t^i, r_t^i >$ will be stored in the replay buffer. Meanwhile, the meta critic collects the history information of task i, i.e., $\mathcal{H}_t = \{(s_k, s_{k+1}, a_k, r_k)\}, k \in [t - \bar{t}, t - 1]$ from the replay buffer together with the state-action pair $< s_t^i, a_t^i >$ to give a task specific Q-value for updating the actor networks. The meta critic is also updated by the trajectories of all tasks in the replay buffer. In the online learning phase, for a newly-coming real-time task, the update of the actor network is the same with that of the meta learning phase while the critic is kept static.

IV. Simulation Results

For the simulation, we set the number of antennas of BS $M = 8$, number of users $K = 4$. We compared it to two benchmark algorithms: 1)Twin delayed deep deterministic policy gradient (TD3) [2] without the meta critic and 2) Zero-Force Exhausting (ZF Exhaust) [1], where the digital beamforming bases on the ZF method, and the RIS phase shift optimization is performed via the exhaustion attack method. We assume that the task is updated every 300 episodes, each of which consists of 20 time slots.

Algorithm 1 MC-DDPG Algorithm

Meta Training:
input: Multiple task samples.
Initialize: (For each task i) Critic Networks and actor network; Target Networks;Replay Buffer;
for *eps* in range(*MaxEpisode*) **do**
  Sample I tasks and initialize states.
  for t in range(*MaxStep*) **do**
    for each task i **do**
      Select action and get reward and next state.
      Store transition tuple into the replay buffer
      Sample a batch of data and update the Meta Critic.
      Update the actor networks with delay.
      Soft-update target networks with delay.
Output: Well-trained meta critic.

Online Training:
input:A new task;Well-trained meta critic.
Initialize: Policy network; Replay Buffer.
for *eps* in range(*MaxEpisode*) **do**
  Initialize system state.
  for t in range(*MaxStep*) **do**
    Select action and get reward and next state.
    Store transition tuple into the replay buffer.
    Sample a batch of data and update the actor network.
Output: Trained policy of actor.

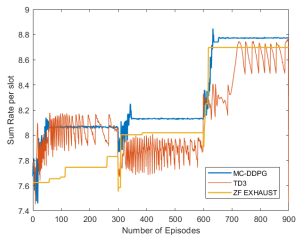

Fig. 3. Sum rate performance with respect to varying users' locations

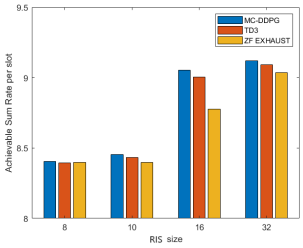

Fig. 4. Achievable sum rate v.s. the number of RIS elements

Fig. 3 shows the performance of the proposed scheme when mobile users move rapidly. The proposed MC-DDPG can rapidly converge to a higher sum rate compared to the benchmarks. As the MC-DDPG can converge within 100 episodes and we set each user's position changes 0.01m each episode, we remark that it can support the user mobility at a minimum speed of 36 km/h. Fig. 4 shows the sum rate varying with the number of RIS elements N. We observe that the proposed MC-DDPG converges to a higher sum rate as N increases from 64 to 1024 compared to two benchmarks, which shows its capability of supporting large-scale RIS.

References

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

[2] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Int. Conf. Machine Learning (ICML)*, Jul. 2018, pp. 1587–1596.