SCIENCE CHINA
Information Sciences

• Supplementary File •

# Beamforming prediction based on the multireward DQN framework for UAV-RIS-assisted THz communication systems

Yuewei WU[1], Peng XU[1*], Yi LV[1], Dongming Wang[2*], Feifei Gao[3*] & Jiangzhou Wang[4*]

[1] *School of Electronics and Information Engineering, Shenyang Aerospace University, Shenyang* 110000*, China;*
[2] *National Mobile Communications Research Laboratory, Southeast University, Nanjing* 210096*, China;*
[3] *Department of Automation, Tsinghua University, Beijing* 100084*, China;*
[4] *School of Engineering, University of Kent, Canterbury* CT2 7NT*, UK*

## Appendix A System and channel model

Here, we consider a terahertz (THz) communication system employing an orthogonal frequency division multiplexing (OFDM) architecture with $K$ subcarriers. The system consists of a base station equipped with $A$ antennas serving as the signal transmitter, and multiple users act as the signal receivers, each of whom is equipped with a single antenna. While some users establish a direct communication link with the base station, others experience signal blockage due to the surrounding structures. To address this challenge, a reconfigurable intelligent surface (RIS) consisting of $M$ elements is deployed on an unmanned aerial vehicle (UAV) fixed at an altitude of 80 meters to reflect signals. Let $\mathbf{H}_k, \mathbf{G}_k \in \mathbb{C}^{M\times 1}$ denote the channels from the BS to the RIS and from the RIS to the user, respectively, at the $k^{th}$ subcarrier. The received signal at the $k^{th}$ subcarrier can be expressed as

$$y_k = \mathbf{H}_k^T \mathbf{\Psi} \mathbf{G}_k^T s_k + z_k = (\mathbf{H}_k \odot \mathbf{G}_k)^T \varphi s_k + z_k, \tag{A1}$$

where $s_k \in \mathbb{C}$ is the signal transmitted over the $k^{th}$ subcarrier, and each subcarrier satisfies the power constraint $E\left[|s_k|^2\right] = \frac{P_k}{K}$, with $P_k$ representing the total transmit power. $z_k \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma^2\right)$ denotes the Gaussian white noise at the $k^{th}$ subcarrier. $\mathbf{\Psi} \in \mathbb{C}^{I\times I}$ denotes the RIS interaction diagonal matrix, i.e., $\mathbf{\Psi} = diag(\varphi_l)$, where $\varphi$ represents the RIS effective phase shift and can be expressed as $[\varphi]_l = e^{j\phi_l}$. All $\varphi$ are selected from a predefined codebook $\mathcal{P}$, which is generated using a uniform planar array (UPA) structure.

In this paper, a broadband geometric THz channel model from [1] is used. To mitigate the beam splitting effect in the broadband model, the model is equipped with a total of $L$ clusters. Each cluster $l \in \{1, \cdots, L\}$ contributes a delay ray from the BS to the RIS (the same applies from the RIS to the user), at which point the frequency domain delay channel vector can be defined as follows: [2]

$$\mathbf{H}_k = \sqrt{\frac{M}{\rho_T}} \sum_{d=0}^{D-1} \sum_{l=1}^{L} \alpha_l \mathbf{a}_{RIS}(\theta_l, \phi_l) p(dT_s - \tau_l) e^{-j\frac{2\pi k}{K}d} \tag{A2}$$

where $\mathbf{a}_{RIS}(\theta_l, \phi_l) \in \mathbb{C}^{M\times 1}$ is the RIS array response vector, and $\theta_l, \phi_l \in [0, 2\pi)$ are the azimuth and elevation angles of arrival, respectively. $\alpha_l \in \mathbb{C}$ is the complex path gain. $\tau_l \in \mathbb{R}$ denotes the pulse shaping function for $T_s$-spaced signaling evaluated at $\tau$ seconds. $\rho_T$ denotes the uplink path loss.

According to the system and channel models described above, the maximum achievable rate at the receiver is defined as

$$R = \max_{\varphi \in \mathbf{P}} \frac{1}{K} \sum_{k=1}^{K} \log_2(1 + \mathrm{SNR}|(\mathbf{H}_k \odot \mathbf{G}_k)^T \varphi|^2) \tag{A3}$$

The maximum power budget assigned to the active RIS can be written as [3]

$$P_{RIS} = \sum_{i=1}^{n} \|\mathbf{\Psi}\mathbf{G}_k\|^2 + \|\mathbf{\Psi}\|^2 \sigma_k^2 \tag{A4}$$

where $\sigma_k$ represents the composite noise containing complex environmental information at the active RIS units.

## Appendix B Simulation setup

This appendix elaborates on the parameter settings of the communication model and the multireward-based double deep Q-network (MRDDQN) proposed in the main text.

[*] Corresponding author (email: xup024@vip.126.com, wangdm@seu.edu.cn, feifeigao@tsinghua.edu.cn, j.z.wang@kent.ac.uk)

1) Parameter settings of the communication model: The DeepMIMO dataset in [2] was used to generate the channels based on the outdoor ray-tracing scenario ”O1-drone”. The dataset parameters are summarized in Table B1. In this scenario, BS 1 remains stationary on the ground, while BS 2 is a flying RIS-RIS attached to a UAV. BS 2 was positioned at an altitude of 80m on the UAV. Additionally, each user can move randomly within a predefined range along the x and y axes.

**Table B1** Parameters of the communication model

| Parameters | Value |
|---|---|
| Center Frequency | 200GHz |
| Active BS | BS 1, BS 2(flying RIS) |
| Active users | From row R235 to row R290 |
| Number of BS 1 antennas | $(M_x,M_y,M_z)$ = (64, 1, 1) |
| Number of BS 2 antennas | $(M_x,M_y,M_z)$ = (256, 1, 1) |
| Bandwidth | 1GHz |
| Number of OFDM subcarriers | 2048 |
| OFDM sampling factor | 1 |
| OFDM limit | 64 |
| BS Antenna spacing | $0.5\lambda$ |

2) Parameter settings of the MRDDQN: States are represented by the sampled channels of each receiver, while actions are depicted by the candidate interaction vector chosen from a predefined codebook $\mathcal{P}$. To mitigate computational complexity, only the initial 64 subcarriers were considered. The neural network architecture comprises four fully-connected layers. The first fully connected layer has input and output dimensions, both set to 512. The second layer is an additional linear layer with the same input and output dimensions as the first. The next module takes an input of size 512 and passes the data flow through a fully connected layer with 256 nodes. Subsequently, the output enters another linear layer with size 1, representing the estimation of the state value. The last module was designed to estimate the action values. Similar to the preceding module, it receives an input of size 512 and passes the data flow through a fully connected layer with 256 nodes. Finally, the output enters another linear layer with an output size of 256. Based on the user units activated in the communication model, a total of 30,000 data samples were generated. Of these, 80% were designated as the training dataset, and 20% were designated as the testing dataset. To manage the training process effectively, this model employs a replay buffer containing all training samples with a batch size of 512. The adaptive learning rate optimization algorithm, Adam, is used to dynamically adjust the learning rates by combining the first- and second-moment estimates of the gradients, and the initial learning rate is set to $2.5 \times 10^{-4}$.

## Appendix C Proposed MRDDQN model

This section describes the framework of the proposed MRDDQN model. The operates in two phases: the learning phase and the prediction phase.

1) Learning phase: As depicted in Algorithm C1, at every coherence block $s$, the learning phase undergoes the following four steps.

- Sampled channel estimation (line 4): According to the channel model described in Appendix A, the matrix $\mathbf{a}_{RIS}$ is designed to select the entries corresponding to the active RIS units, where $\mathbf{a}_{RIS}$ is an $\bar{M} \times M$ selection matrix. The sampled channel vector from the transmitter/receiver to the active RIS elements, $\bar{\mathbf{H}}_k, \bar{\mathbf{G}}_k \in \mathbb{C}^{\bar{M}\times 1}$, can be expressed as $\bar{\mathbf{H}}_k = \mathbf{a}_{RIS}\mathbf{H}_k$ and $\bar{\mathbf{G}}_k = \mathbf{a}_{RIS}\mathbf{G}_k$. Then, the overall RIS sampled channel vector can be expressed as $\bar{\mathbf{h}}_k = \bar{\mathbf{H}}_k \odot \bar{\mathbf{G}}_k$ and the concatenated channel vector is defined as $\bar{\mathbf{h}} = vec([\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, ..., \bar{\mathbf{h}}_K])$. Finally, let $\bar{\mathbf{h}}(s)$ denote the concatenated sampled channel vector at the $s^{th}$ coherence block, where $s \in \{1, \cdot\cdot\cdot, S\}$ and $S$ is the total number of data samples used to construct the learning dataset. For every channel coherence block $s$, the transmitter and receiver transmit two orthogonal uplink pilots. The active RIS units will receive these pilots and estimate the sampled channel vectors to construct the multipath signature, which is expressed as

$$\hat{\bar{\mathbf{H}}}_k(s) = \bar{\mathbf{H}}_k(s) + \mathbf{v}_k, \hat{\bar{\mathbf{G}}}_k(s) = \bar{\mathbf{G}}_k(s) + \mathbf{w}_k, \tag{C1}$$

$$\hat{\bar{\mathbf{h}}}_k(s) = \hat{\bar{\mathbf{H}}}_k(s) \odot \hat{\bar{\mathbf{G}}}_k(s), \tag{C2}$$

$$\hat{\bar{\mathbf{h}}}(s) = vec\left(\left[\hat{\bar{\mathbf{h}}}_1(s), \hat{\bar{\mathbf{h}}}_2(s), ..., \hat{\bar{\mathbf{h}}}_K(s)\right]\right), \tag{C3}$$

where $\mathbf{v}_k, \mathbf{w}_k \sim \mathcal{N}_{\mathbb{C}}\left(0, {\sigma_n}^2\mathbf{I}\right)$ are the received noise vectors at the active RIS units.

- Exhaustive beam training (lines 5-9): In this step, the RIS performs an exhaustive search over the reflection codewords from the reflection codebook $\mathcal{P}$. Specifically, the RIS tries out every candidate reflection beamforming vector, $\varphi_n$, $n = 1, ..., |\mathcal{P}|$, and active RIS units receive feedback from the user indicating the achievable rate and the power budget of the active RIS attained by using this reflection beamforming vector.

- Experience construction (lines 10-12): After receiving the achievable rate and the power budget, the Q-function evaluates them and calculates the corresponding reward value according to the double-reward DDQN. Upon determining the optimal RIS beamforming vector for each pertinent block $s$, the associated environmental label, best RIS beamforming vector, and reward value are stored as priority experiences in the replay buffer $\mathcal{D}$, constituting the $\left\langle s, a, r, s' \right\rangle$ action tuple.

- Model training (lines 14-23): For each epoch, the model retrieves action pairs from the replay buffer $\mathcal{D}$, extracting an amount equal to the batch size. First, the model obtains the Q-value of the optimal action according to the current state. It then obtains the optimal action of the next state through double-Q learning and calculates the Q-update of this action. Subsequently, the Adam optimizer is utilized to update the gradient of the loss function between the Q-value and the Q-update until convergence is achieved. It learns how to map an input state to an output action.

2) Prediction phase: During the prediction phase, the trained model is saved and switched to the evaluation mode, in which it utilizes the acquired sampled channel vectors of the current state to predict the optimal RIS beamforming vector of the next state. This phase comprises the following two steps.

- Sampled channel estimation (line 26): This step is the same as the first step in the learning phase. The active RIS units receive uplink pilots to estimate and construct the concatenated sampled channel vector $\hat{\hat{\mathbf{h}}}$.
- Optimal beamforming vector prediction (lines 27 and 28): In this step, the trained model predicts the Q-value, which represents the best RIS beamforming vector.

**Algorithm C1** Double-reward-based DDQN for RIS Beamforming Prediction

```
 1: Phase 1 : Learning phase
 2: Initialization: Policy network $Q(s,a|\theta)$, target network $Q^*(s,a|\theta)$, replay buffer $\mathcal{D}$
 3: for $s = 1$ to $S$ do
 4:     active RIS receives two pilots to estimate $\hat{\hat{\mathbf{h}}}(s)$;
 5:     for $n = 1$ to $\mathcal{P}$ do
 6:         active RIS reflects using $\varphi_n$ beam.;
 7:         active RIS receives the feedback $R(n), P_{RIS}(n)$;
 8:         active RIS quantizes the reward $R_{beam}(n)$;
 9:     end for
10:     active RIS receives two pilots to estimate $\hat{\hat{\mathbf{h}}}(s+1)$;
11:     $\left\langle s,a,r,s' \right\rangle \leftarrow \left\langle \hat{\hat{\mathbf{h}}}(s), \varphi(s), R_{beam}(s), \hat{\hat{\mathbf{h}}}(s+1) \right\rangle$;
12:     Store the experience$\left\langle s,a,r,s' \right\rangle$in $\mathcal{D}$;
13: end for
14: for every epoch do
15:     Minibatch experiences from $\mathcal{D}$ for training;
16:     Feedforward $s$ to calculate $Q(s,a|\theta)$;
17:     Feedforward $s'$ to calculate $Q^*\left(s',a'|\theta\right)$;
18:     Feedforward $a'$ to find $Q\left(s,a'|\theta\right)$;
19:     Predict and calculate $Q^*\left(s',a^*|\theta\right) \leftarrow R_{beam} + \gamma Q^*\left(s',a'|\theta\right)$;
20:     Use Huber loss function to calculate $Loss\left\{Q\left(s,a'|\theta\right), Q^*\left(s',a^*|\theta\right)\right\}$;
21:     Use Adam optimizer to update gradients;
22:     $s \leftarrow s'$;
23:     Until reaching a terminal goal;
24: end for
25: Phase 2 : Prediction phase
26: active RIS receives two pilots to estimate $\hat{\hat{\mathbf{h}}}$;
27: Predict the best action using the trained model;
28: active RIS reflects using $\varphi^*$.
```

## Appendix D Model training design

This section provides supplementary explanations of the input and loss function settings of the MRDDQN.

1) Input representation: The sampled channel vector serves as the input for the deep Q-network. To ensure uniformity across the datasets, all samples were normalized by the maximum absolute value of the entire input dataset [4]. This approach maintains the encoded distance information within the multipath signatures. Each complex entry within the input data is decomposed into its real and imaginary components, effectively doubling the dimensionality of each input vector to 2KM.

2) Training loss function: According to the Q-value function, the loss function of the MRDDQN should be updated as follows:

$$L_i(\theta_i) = E_{s,a,r,s'} \sim U(D)\left[\left(y_i - Q_{R_{beam}}(s,a;\theta_i)\right)^2\right], \tag{D1}$$

with

$$y_i = R_k\left(s,a,s'\right) + \gamma Q_{R_{beam}}\left(s', \arg\max_{a'} Q_{action}\left(s',a';\theta_i\right);\theta_i^-\right)), \tag{D2}$$

where $a'$ is taken from $\theta_i$ and the value is taken from $\theta_i^-$. Both $\theta_i$ and $\theta_i^-$ are a set of parameters utilized in computing the target network, where the former evaluates action selection and the latter evaluates state value [5]. Huber loss is used in the proposed model to minimize the loss function.

**References**

1 Taha A, Alrabeiah M, Alkhateeb A. Enabling Large Intelligent Surfaces with Compressive Sensing and Deep Learning, IEEE Access, 2021, 99: 1-1

2 Abuzainab N, Alrabeiah M, Alkhateeb A, et al. Deep Learning for THz Drones with Flying Intelligent Surfaces: Beam and Handoff Prediction, In: Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 2021. 1-6

3 Farrag S, Maher E A, El-Mahdy A, et al. Sum Rate Maximization of Uplink Active RIS and UAV-assisted THz Mobile Communications, In: Proceedings of 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN), Vilanova i la Geltru, Spain, 2023. 1-7

4 Zhang Y, Alrabeiah M, Alkhateeb A, Deep Learning for Massive MIMO with 1-Bit ADCs: When More Antennas Need Fewer Pilots, IEEE Wireless Communications Letters, 2020, 9:1273-1277

5 Mnih v, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning, Nature, 2015, 518:529-533H