Reinforcement Learning for Beam Pattern Design in Millimeter Wave and Massive MIMO Systems

Yu Zhang, Muhammad Alrabeiah, and Ahmed Alkhateeb

2020 54th Asilomar Conference on Signals, Systems, and Computers | 978-0-7381-3126-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF51394.2020.9443430

Abstract—**Deploying large scale antenna arrays is a key characteristic of current and future wireless communication systems. However, due to some non-ideal practical conditions, such as the unknown array geometry or possible hardware impairments, the accurate channel state information becomes hard to acquire. This impedes the design of beamforming/combining vectors that are crucial to fully exploit the potential of the large-scale MIMO systems or to combat the high path-loss in millimeter wave (mmWave) communications. In this paper, we propose a novel solution that leverages deep reinforcement learning (DRL) to learn the beam pattern that is optimized for a group of users without the explicit knowledge of the channels. Simulation results show that the developed solution is capable of finding the near optimal beam pattern with quantized phase shifters and with only requiring the beamforming gain feedback from the users.**

I. Introduction

Leveraging the large bandwidth available at millimeter wave (mmWave) frequency bands requires the deployment of large antenna arrays. However, to balance the overall hardware cost, cheap and low-precision radio components might be adopted. This leads to some non-ideal practical conditions, such as unknown array geometry or possible hardware impairments. In this situation, the performance of the commonly used beams (such as the ones in classical beamsteering codebooks) degrades drastically due to their unawareness of the environment and hardware. Furthermore, the accurate channel state information is generally hard/prohibitive to estimate due to the possible hardware impairments and the large number of antennas. As a result, classical or data-driven beam pattern/codebook design approaches, e.g. [1], may not be feasible.

Prior Work: Designing beamforming codebooks is a key step in realizing the potential of mmWave MIMO communications, and it has been an important research topic for quite some time [2]–[5]. With large-scale MIMO systems, the hardware limitations (especially at mmWave/THz) and the use of analog-only or hybrid transceiver architectures impose new constraints on the codebook design problems. This has motivated the development of new beamforming codebooks with single-lobe and narrow beams [6]. Although very directive, those codebooks bring with them an increased training overhead. As such, [7] and [8] has explored hierarchical codebook structures, which implements different levels of beam widths.

Contribution: In this paper, we propose a deep reinforcement learning (DRL) based solution to learn the optimized beam pattern for a group of users. This is done by utilizing a novel Wolpertinger architecture [9] which is designed to efficiently explore the large discrete action space. The proposed model accounts for key hardware constraints such as the phase-only, constant-modulus, and quantized-angle constraints [10]. This is realized by defining the state directly as the phases of the analog phase shifters and the action as the change of phases within the quantized phase set. Simulation results show that the proposed solution is capable of finding the near optimal beam pattern and achieving a beamforming gain compared to that of equal gain combining.

II. System and Channel Models

In this section, we introduce in detail our adopted system and channel models. We also describe how the model considers arbitrary array geometries with possible hardware impairments.

A. System Model

We consider the system model where a mmWave massive MIMO base station (BS) with M antennas is communicating with a single-antenna user. Further, given the high cost and power consumption of mixed-signal components, we consider a practical system where the BS has only one radio frequency (RF) chain and employs analog-only beamforming using a network of r-bit quantized phase shifters. Therefore, the beamforming vector can be written as

$$\mathbf{w} = \frac{1}{\sqrt{M}} \left[e^{j\theta_1}, e^{j\theta_2}, \dots, e^{j\theta_M}\right]^T, \tag{1}$$

where each phase shift θ_m is selected from a finite set $\boldsymbol{\Theta}$ with 2^r possible discrete values drawn uniformly from $(-\pi, \pi]$. In the uplink transmission, if a user u transmits a symbol $x \in \mathbb{C}$ to the base station, where the transmitted symbol satisfies the average power constraint $\mathbb{E}\left[|x|^2\right] = P_x$, the received signal at the base station after combining can be expressed as

$$y_u = \mathbf{w}^H \mathbf{h}_u x + \mathbf{w}^H \mathbf{n}, \tag{2}$$

where $\mathbf{h}_u \in \mathbb{C}^{M\times 1}$ is the uplink channel vector between the user u and the base station antennas and $\mathbf{n} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sigma_n^2 \mathbf{I}\right)$ is the receive noise vector at the base station.

B. Channel Model

We adopt a general geometric channel model for $\mathbf{h}_u$. Assume that the signal propagation between the user u and the base station consists of L paths. Each path ℓ has a complex

Yu Zhang, Muhammad Alrabeiah and Ahmed Alkhateeb are with Arizona State University (Email: y.zhang, malrabei, alkhateeb@asu.edu). This work is supported by the National Science Foundation under Grant No. 1923676.

gain α_ℓ and an angle of arrival ϕ_ℓ. Then, the channel vector can be written as

$$\mathbf{h}_u = \sum_{\ell=1}^{L} \alpha_\ell \mathbf{a}(\phi_\ell), \tag{3}$$

where $\mathbf{a}(\phi_\ell)$ is the array response vector of the base station. The definition of $\mathbf{a}(\phi_\ell)$ depends on the array geometry and hardware impairments. Next, we discuss that in more detail.

C. Hardware Impairments Model

Most of the prior work on mmWave signal processing has assumed uniform antenna arrays with perfect calibration and ideal hardware [3], [6], [8], [10]. In this paper, we consider a more general antenna array model that accounts for arbitrary geometry and hardware impairments, and target learning beam pattern that mitigates the influence of those unknown factors. While the beam pattern learning solution that we develop in this paper is general for various kinds of array geometries and hardware impairments, we evaluate the proposed solution in Section V with respect to two main characteristics of interest, namely non-uniform spacing and phase mismatch between the antenna elements. For linear arrays, the array response vector can be modeled to capture these characteristics as follows

$$\mathbf{a}(\phi_\ell) = \Big[e^{j(kd_1\cos(\phi_\ell)+\Delta\theta_1)}, e^{j(kd_2\cos(\phi_\ell)+\Delta\theta_2)}, \dots, e^{j(kd_M\cos(\phi_\ell)+\Delta\theta_M)}\Big]^T, \tag{4}$$

where d_m is the position of the m-th antenna, and $\Delta\theta_m$ is the additional phase shift incurred at the m-th antenna (to model the phase mismatch). Without loss of generality, we assume that d_m and $\Delta\theta_m$ are fixed yet unknown random realizations drawn from the distributions $\mathcal{N}\left((m-1)d, \sigma_d^2\right)$ and $\mathcal{N}\left(0, \sigma_p^2\right)$ respectively, where d is the ideal antenna spacing, σ_d and σ_p model the standard deviations of the random antenna position and phase mismatch. Besides, we impose an additional constraint $d_1 < d_2 < \cdots < d_M$ to make sure the generated antenna positions physically meaningful.

III. Problem Definition

In this paper, we investigate the beam pattern design problem for mmWave and massive MIMO system with unknown array geometry and hardware impairment. Given the system and channel models described in Section II, the SNR after combining for user u can be written as

$$\mathsf{SNR}_u = \frac{\left|\mathbf{w}^H\mathbf{h}_u\right|^2}{\|\mathbf{w}\|^2}\rho = \left|\mathbf{w}^H\mathbf{h}_u\right|^2\rho, \tag{5}$$

where $\|\mathbf{w}\|^2 = 1$ is implicitly used and $\rho = \frac{P_x}{\sigma_n^2}$. Besides, we define the beamforming/combining gain of adopting $\mathbf{w}$ as a transmit/receive beamformer for user u as

$$g_u = \left|\mathbf{w}^H\mathbf{h}_u\right|^2. \tag{6}$$

It can be seen that maximizing (6) is equivalent to maximizing the SNR in (5). Therefore, the objective of this paper is to design (learn) the beamforming vector $\mathbf{w}$ that maximizes the beamforming/combining gain given by (6) **averaged over the set of the users with similar channels.** Therefore, the beam pattern learning problem can be formulated as

$$\mathbf{w}_{\mathsf{opt}} = \arg\max_{\mathbf{w}} \frac{1}{|\boldsymbol{\mathcal{H}}|}\sum_{\mathbf{h}_u\in\boldsymbol{\mathcal{H}}} \left|\mathbf{w}^H\mathbf{h}_u\right|^2, \tag{7}$$

$$\text{s.t.}\ w_m = \frac{1}{\sqrt{M}}e^{j\theta_m},\ \forall m = 1, ..., M, \tag{8}$$

$$\theta_m \in \boldsymbol{\Theta},\ \forall m = 1, ..., M, \tag{9}$$

where w_m is the m-th element of the beamforming vector and $\boldsymbol{\mathcal{H}}$ the channel set that is supposed to contain a single channel or multiple similar channels. It is worth mentioning that the constraint in (8) is imposed to uphold the adopted analog-only system model, and the constraint in (9) is to respect the quantized phase-shifters hardware constraint.

Due to the unknown array geometry as well as possible hardware impairments, the accurate channel state information is generally hard to acquire. This means that all the channels $\mathbf{h}_u \in \boldsymbol{\mathcal{H}}$ in the objective function are possibly unknown. Instead, the base station may only have access to the beamforming/combining gain g_u (or equivalently the Received Signal Strength Indicator (RSSI)) reported by each user. Therefore, problem (7) is hard to solve in a general sense for the unknown parameters in the objective function as well as the non-convex constraint (8) and the discrete constraint (9). Given that **this problem is essentially a search problem with feedbacks in a dauntingly huge yet finite and discrete space,** we consider leveraging the powerful exploration capability of deep reinforcement learning to efficiently search over the space to find the optimal or near-optimal solution.

IV. Beam Pattern Learning

In this section, we present our proposed DRL-based algorithm for addressing the beam pattern design problem (7). It is worth mentioning that when viewing the problem from a reinforcement learning perspective, it features a **finite yet very high dimensional** action space. This makes the traditional learning frameworks (such as deep Q-learning, deep deterministic policy gradient, etc.) hard to apply. Therefore, we adopt a novel architecture called Wolpertinger to enable the efficient search in a large discrete action space, the details of which can be found at [9].

1) Reinforcement Learning Setup: To solve the problem with reinforcement learning, we first specify the corresponding building blocks of the learning algorithm as follows:

- **State:** We define the state $\mathbf{s}_t$ as a vector that consists of the phases of all the phase shifters at the t-th iteration, that is, $\mathbf{s}_t = [\theta_1, \theta_2, \dots, \theta_M]^T$. This phase vector can be converted to the actual beamforming vector by applying (1). Since all the phases in $\mathbf{s}_t$ are selected from $\boldsymbol{\Theta}$, and all the phase values in $\boldsymbol{\Theta}$ are within $(-\pi, \pi]$, (1) essentially defines a bijective mapping from the phase vector to the beamforming vector. Therefore, for simplicity, we will use the term “beamforming vector” to refer to both this phase vector and the actual beamforming vector (the conversion is given by (1)), according to the context.

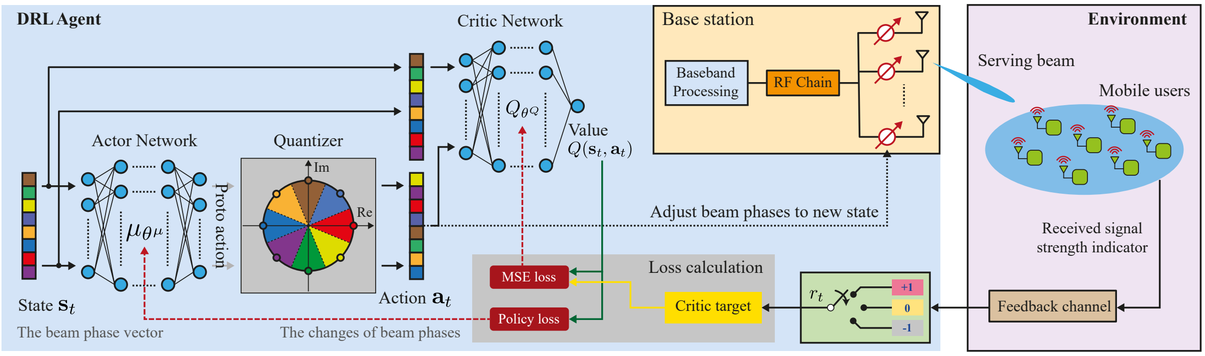


Fig. 1. The proposed beam pattern design framework with deep reinforcement learning. The schematic shows the agent architecture, and the way it interacts with the environment.

- **Action:** We define the action $\mathbf{a}_t$ as the element-wise changes to all the phases in $\mathbf{s}_t$. Since the phases can only take values in $\mathbf{\Theta}$, a change of a phase means that the phase shifter selects a value from $\mathbf{\Theta}$. Therefore, the action is directly specified as the next state, i.e. $\mathbf{s}_{t+1} = \mathbf{a}_t$.
- **Reward:** We define a ternary reward mechanism, i.e. the reward r_t takes values from $\{+1, 0, -1\}$. We compare the beamforming gain achieved by the current beamforming vector, denoted by g_t, with two values: (i) an adaptive threshold β_t, and (ii) the previous beamforming gain g_{t-1}. The reward is computed using the following rule
  - $g_t \geq \beta_t$, $r_t = +1$;
  - $g_t < \beta_t$ and $g_t \geq g_{t-1}$, $r_t = 0$;
  - $g_t < \beta_t$ and $g_t < g_{t-1}$, $r_t = -1$.

It is important to note that the adopted adaptive threshold mechanism does not rely on any prior knowledge of the channel distribution. The threshold value starts from zero and whenever the BS tries a new beam and the resulting beamforming gain surpasses the current threshold, the system updates the threshold by the value of this new beamforming gain. Besides, since the update of threshold also marks a successful detection of a new beam that achieves the best beamforming gain so far, the BS also records this beamforming vector. As can be seen in the reward definition, in order to calculate the reward, the system always tracks two quantities, which are the previous beamforming gain and the best beamforming gain achieved so far (i.e. the threshold).

2) Environment Interaction: As mentioned in Sections I and III, due to the possible hardware impairments, accurate channel state information is generally unavailable. Therefore, the base station can only resort to the beamforming gain feedback reported by the users to adjust its beam pattern in order to achieve a better performance. Upon forming a new beam $\tilde{\mathbf{w}}$, the base station transmits a pilot x by using this beam and gets feedback from every user. Then, it averages all the beamforming gain feedbacks

$$\bar{g} = \frac{1}{|\boldsymbol{\mathcal{H}}|} \sum_{\mathbf{h}_u \in \boldsymbol{\mathcal{H}}} \left| \tilde{\mathbf{w}}^H \mathbf{h}_u \right|^2, \qquad (10)$$

where $\boldsymbol{\mathcal{H}}$ represents the targeted user channel set. Recall that (10) is the same as evaluating the objective function of (7) with the current beamforming vector $\tilde{\mathbf{w}}$. Depending on whether or not the new average beamforming gain surpasses the previous one as well as the current threshold, the base station gets either reward or penalty, based on which it can judge the "quality" of the current beam and decide how to move.

Algorithm 1 DRL Based Beam Pattern Learning

```
Initialize actor network $\mu(\mathbf{s}|\theta^\mu)$ and critic network $Q(\mathbf{s}, \mathbf{a}|\theta^Q)$ with random weights $\theta^\mu$ and $\theta^Q$
Initialize target networks $\mu'$ and $Q'$ with the weights of actor and critic networks' $\theta^{\mu'} \leftarrow \theta^\mu$ and $\theta^{Q'} \leftarrow \theta^Q$
Initialize the replay memory $\mathcal{D}$, minibatch size $B$
Initialize adaptive threshold $\beta = 0$ and the previous average beamforming gain $g_1 = 0$
Initialize a random process $\mathcal{N}$ for action exploration
Initialize a random beamforming vector $\mathbf{w}_1$ as the initial state $\mathbf{s}_1$
for $t = 1$ to $T$ do
    Receive a predicted action from actor network with exploration noise $\widehat{\mathbf{a}}_t = \mu(\mathbf{s}_t|\theta^\mu) + \mathcal{N}_t$
    Quantize the predicted action to a valid beamforming vector $\mathbf{a}_t$ according to (11)
    Execute action $\mathbf{a}_t$, observe reward $r_t$ and update state to $\mathbf{s}_{t+1} = \mathbf{a}_t$
    Update the threshold $\beta$ and the previous beamforming gain $g_t$
    Store the transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in $\mathcal{D}$
    Sample a random mini batch of $B$ transitions $(\mathbf{s}_b, \mathbf{a}_b, r_b, \mathbf{s}_{b+1})$ from $\mathcal{D}$
    Calculate target $y_b = r_b + \gamma Q'(\mathbf{s}_{b+1}, \mu'(\mathbf{s}_{b+1}|\theta^{\mu'})|\theta^{Q'})$
    Update the critic network by minimizing the mean squared loss $L = \frac{1}{B}\sum_b (y_b - Q(\mathbf{s}_b, \mathbf{a}_b|\theta^Q))^2$
    Update the actor network using the sampled policy gradient given by
    $-\frac{1}{B}\sum_{b=1}^{B} \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})|_{\mathbf{s}=\mathbf{s}_b, \mathbf{a}=\mu(\mathbf{s}_b|\theta^\mu)} \nabla_{\theta^\mu} \mu(\mathbf{s}|\theta^\mu)|_{\mathbf{s}=\mathbf{s}_b}$
    Update the target networks every $C$ iterations
end for
```

3) Exploration: The exploration happens after the actor network predicts the action $\widehat{\mathbf{a}}_{t+1}$ based on the current state (beam) $\mathbf{s}_t$. Upon obtaining the predicted action, an additive

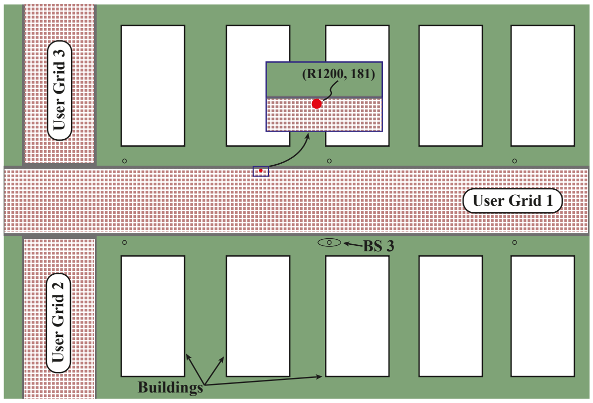


Fig. 2. The top view of the considered communication scenario.

noise is added element-wisely to $\widehat{\mathbf{a}}_{t+1}$ for the purpose of exploration, which is a customary way in the context of reinforcement learning with continuous action spaces [11], [12]. In our problem, we use temporally correlated noise samples generated by an Ornstein-Uhlenbeck process [13], which is also used in [9]. It is worth mentioning that a proper configuration of the noise generation parameters has significant impact on the learning process. Normally, the extent of exploration (noise power) is set to be a decreasing function with respect to the iteration number, which is commonly known as exploration-exploitation tradeoff [11]. Furthermore, the exact configuration of noise power should relate to specific applications. In our problem, for example, the noise is directly added to the predicted phases. Thus, at the very beginning, the noise should be strong enough to perturb the predicted phase to any other phases in $\mathbf{\Theta}$. By contrast, when the learning process approaches to the termination (the learned beam already performs well), the noise power should be decreased to a smaller level that is only capable of perturbing the predicted phase to its adjacent phases in $\mathbf{\Theta}$.

4) Quantization: The predicted beam (with exploration noise added) should be quantized in order to be a valid new beam that can be implemented by the discrete phase-shifters. Therefore, each quantized phase in the new vector can be calculated as

$$[\mathbf{s}_{t+1}]_m = \underset{\theta \in \mathbf{\Theta}}{\arg\min} \left| \theta - [\widehat{\mathbf{s}}_{t+1}]_m \right|, \forall m = 1, 2, \dots, M, \quad (11)$$

which is essentially a nearest neighbor lookup (i.e. a KNN classifier with $k = 1$).

5) Forward Computation and Backward Update: The current state $\mathbf{s}_t$ and the new state $\mathbf{s}_{t+1}$ (recall that we directly set $\mathbf{s}_{t+1} = \mathbf{a}_t$) are then fed into the critic network to compute the Q value, based on which the targets of both actor and critic networks are calculated. This completes a forward pass. Following that, a backward update is performed to the parameters of the actor and critic networks. A pseudo code of the algorithm can be found in Algorithm 1.

V. Simulation Results

In this section, we evaluate the performance of the proposed solution. We first describe the adopted scenario and dataset used in our simulations and then discuss the results.

A. Scenario and Dataset

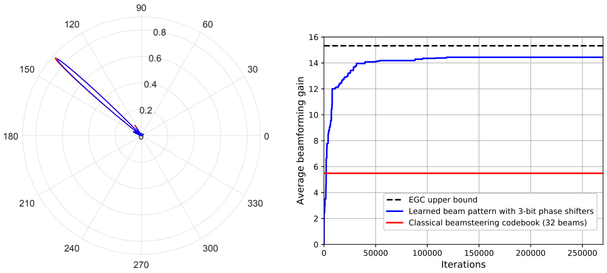
In our simulations, we consider the outdoor scenario 'O1_60' which is offered by the DeepMIMO dataset [14] and is generated based on the accurate 3D ray-tracing simulator Wireless InSite [15]. This scenario comprises two streets and one intersection with three uniform x-y user grids, as shown in Fig. 2. To generate the channels from the users to the base station, we adopt the following DeepMIMO parameters: (1) Scenario name: O1_60, (2) Active BSs: 3, (3) Active users: Row 1200 to 1200, (4) Number of BS antennas in (x, y, z): (1, 32, 1), (5) System bandwidth: 1 GHz, (6) Number of OFDM sub-carriers: 1 (single-carrier), (7) Number of multipaths: 5. From the generated dataset, we further select the user at row 1200 and column 181 in the scenario. The locations of both the selected user and the base station are marked in Fig. 2.

B. Performance Evaluation

We first evaluate our proposed DRL-based beam pattern learning solution on learning a single beam that serves a single user with LOS connection to the base station. As shown in Fig. 3 (a), the proposed solution is capable of finding where the user is and forming a pointed beam to serve that user. By comparing the beam patterns of the equal gain combining/beamforming vector (plotted in red) and the learned beam (plotted in blue), it is evident that the proposed solution can capture the main lobe of the equal gain combining/beamforming vector very well, which explains the excellent performance it achieves. The slight mismatching is mainly due to the use of quantized phase shifters. With 3-bit resolution, each phase shifter can only realize 8 different values of phase shifts drawn uniformly from $(-\pi, \pi]$.

We also compare the performance of the learned single beam with a 32-beam classical beamsteering codebook, illustrated in Fig. 3 (b). As it is commonly known, classical beamsteering codebook normally performs very well in LOS scenario. However, our proposed method achieves higher beamforming gain than the classical beamsteering codebook with negligible iterations. More interestingly, with less than 5×10^4 iterations, the proposed solution can reach more than 90% of the equal gain combining (EGC) upper bound. It is worth mentioning that the EGC upper bound can only be reached when the user's channel is completely known and unquantized phase shifters are deployed. By contrast, our proposed solution can finally achieve almost 95% of the EGC upper bound with 3-bit phase shifters and without any channel information.

The proposed beam pattern learning solution is also evaluated on a system where hardware impairments exist (with the same user considered above). This is a more realistic and interesting scenario, for mmWave systems are susceptible to perturbations like antenna spacing mismatch and phase mismatch. The wavelength in mmWave bands is so small that even slight mismatching can lead to a drastic degradation of the performance. This for sure calls for an intelligent design process that is capable of adapting the beam pattern to the



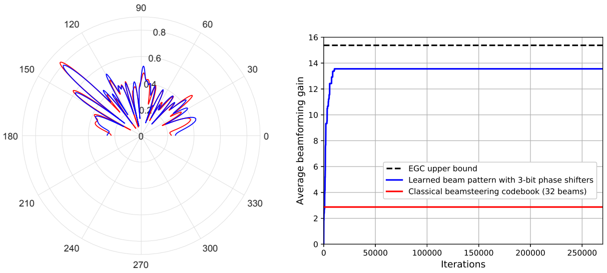
(a) (b)

Fig. 3. The beam pattern learned for a single user with LOS connection to the base station. The base station employs a perfect uniform linear array with 32 antennas and 3-bit phase shifters. (a) shows the beam patterns for the equal gain combining/beamforming vector (red) and the learned beam (blue). (b) shows the learning process.

hardware, mitigating the loss caused by hardware mismatches. The simulation results confirm that our proposed solution is competent to learn such optimized beam pattern for a system with hardware impairments. Fig. 4 (a) shows the beam patterns for both equal gain combining/beamforming vector and the learned beam. At the first glance, the learned beam appears distorted and has multiple low-gain lobes. However, the performance of such beam is excellent. This can be explained by comparing the beam patterns of the learned beam and the equal gain combining/beamforming vector. **As can be seen from the learned beam patterns, our proposed solution intelligently approximates the optimal beam, where all the dominant lobes are well captured.** By contrast, the classical beamsteering codebook fails when the hardware is not perfect, as depicted in Fig. 4 (b). This is because the distorted array pattern incurred by the hardware impairment makes the pointed classical beamsteering codebook beams only able to capture a small portion of the energy, which further results in an inferior beamforming gain. The learned beam shown in Fig. 4 (a) is capable of achieving more than 90% of the EGC upper bound with approximately only 10^4 iterations, as shown in Fig. 4 (b). This is especially interesting for the fact that the proposed solution does not rely on any channel state information. As it is known, the channel estimation in this case relies first on a full calibration of the hardware, which is a hard and expensive process.

VI. Conclusions and Discussions

In this paper, we developed a DRL-based approach to learn the optimized beam pattern for a single user or a group of users with similar channels in mmWave massive MIMO systems. More specifically, we adopt a novel Wolpertinger architecture which is designed to efficiently explore the large discrete action space. The proposed learning framework respects key hardware constraints such as the phase-only, constant-modulus, and quantized-angle constraints. Simulation results show that the proposed solution is capable of finding the near optimal beam pattern which achieves a beamforming gain compared to that of equal gain combining.



(a) (b)

Fig. 4. The beam pattern learned for a single user with LOS connection to the base station. The base station employs a uniform linear array with 32 antennas and 3-bit phase shifters, where hardware impairments exist. The standard deviation of the antenna position is 0.1λ and the standard deviation of the phase mismatches is 0.32π. (a) shows the beam patterns for the equal gain combining/beamforming vector (red) and the learned beam (blue). A transformation of $\sqrt[4]{\cdot}$ is used to better show the finer structure of the beams. (b) shows the learning process.

References

[1] D. Love and R. Heath Jr, "Equal gain transmission in multiple-input multiple-output wireless systems," *IEEE Transactions on Communications*, vol. 51, no. 7, pp. 1102–1110, 2003.

[2] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE Journal on Selected Areas in Commun.*, vol. 26, no. 8, pp. 1341–1365, Oct. 2008.

[3] A. Alkhateeb, O. El Ayach, G. Leus, and R. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[4] J. Mo, B. L. Ng, S. Chang, P. Huang, M. N. Kulkarni, A. Alammouri, J. C. Zhang, J. Lee, and W. Choi, "Beam Codebook Design for 5G mmWave Terminals," *IEEE Access*, vol. 7, pp. 98 387–98 404, 2019.

[5] M. Alrabeiah, Y. Zhang, and A. Alkhateeb, "Neural Networks Based Beam Codebooks: Learning mmWave Massive MIMO Beams that Adapt to Deployment and Hardware," 2020.

[6] J. Wang, Z. Lan, C. Pyo, T. Baykas, C. Sum, M. Rahman, J. Gao, R. Funada, F. Kojima, H. Harada *et al.*, "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1390–1399, Nov. 2009.

[7] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, 2014.

[8] S. Hur, T. Kim, D. Love, J. Krogmeier, T. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, Oct. 2013.

[9] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, and B. Coppin, "Deep reinforcement learning in large discrete action spaces," 2015.

[10] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. Heath, "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Communications Magazine,*, vol. 52, no. 12, pp. 122–131, Dec. 2014.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015.

[13] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Phys. Rev.*, vol. 36, pp. 823–841, Sep 1930. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.36.823

[14] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb 2019, pp. 1–8.

[15] Remcom, "Wireless insite," http://www.remcom.com/wireless-insite.