Orbit correction based on improved reinforcement learning algorithm

Xiaolong Chen, Yongzhi Jia, Xin Qi,[*] Zhijun Wang,[†] and Yuan He
Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, People's Republic of China
School of Nuclear Science and Technology, University of Chinese Academy of Sciences,
Beijing 100049, People's Republic of China

(Received 30 August 2022; accepted 24 March 2023; published 13 April 2023)

Recently, reinforcement learning (RL) algorithms have been applied to a wide range of control problems in accelerator commissioning. In order to achieve efficient and fast control, these algorithms need to be highly efficient, so as to minimize the online training time. In this paper, we incorporated the beam position monitor trend into the observation space of the twin delayed deep deterministic policy gradient (TD3) algorithm and trained two different structure agents, one based on physical prior knowledge and the other using the original TD3 network architecture. Both of the agents exhibit strong robustness in the simulated environment. The effectiveness of the agent based on physical prior knowledge has been validated in a real accelerator. Results show that the agent can overcome the difference between simulated and real accelerator environments. Once the training is completed in the simulated environment, the agent can be directly applied to the real accelerator without any online training process. The RL agent is deployed to the medium energy beam transport section of China Accelerator Facility for Superheavy Elements. Fast and automatic orbit correction is being tested with up to ten degrees of freedom. The experimental results show that the agents can correct the orbit to within 1 mm. Moreover, due to the strong robustness of the agent, when a trained agent is applied to different lattices of different particles, the orbit correction can still be completed. Since there are no online data collection and training processes, all online corrections are done within 30 s. This paper shows that, as long as the robustness of the RL algorithm is sufficient, the offline learning agents can be directly applied to online correction, which will greatly improve the efficiency of orbit correction. Such an approach to RL may find promising applications in other areas of accelerator commissioning.

DOI: 10.1103/PhysRevAccelBeams.26.044601

I. INTRODUCTION

In the particle accelerator, the particles usually deviate from the ideal orbit due to errors of the dipole field and alignment errors of the magnets, resulting in orbital distortion, which leads to degradation of machine performance and even failure. Orbit correction is the most basic step of accelerator beam adjustment and is also one of the most widely studied procedures in accelerator operation. At present, the commonly used orbit correction methods are based on MICADO [1] and singular value decomposition (SVD) [2,3]. The core of these methods is the response matrix, where the corresponding corrector strength is calculated through the response matrix and the beam position monitor (BPM) data. Although these methods have the advantage of being conceptually very simple, they have several important limitations as the number of parameters to control increases [4], such as the necessary and time-consuming remeasurements of the response matrix, retuning of the controller parameters, and increased control imprecisions due to inaccurate measurements of the response, etc. For SVD, in our experience, as long as the beam or lattice is changed, its response matrix needs to be remeasured, which generally takes 4–5 min with the same degrees of freedom. A time-efficient and universally applicable orbit correction method can significantly reduce the time required for beam-line adjustments after each beam replacement.

Recently, investigation of reinforcement learning (RL) for certain accelerator control problems has become increasingly important. Many laboratories have studied RL algorithms already for various control problems, such as orbit correction, beam stability, longitudinal phase space manipulation, etc. [5–8]. The idea of using machine learning for more efficient orbit correction first emerged in the 1990s [9], but early attempts were limited to the development of computer performance and machine

[*] Corresponding author.
qixin2002@impcas.ac.cn

[†] Corresponding author.
wangzj@impcas.ac.cn

learning itself. Therefore, many relevant achievements have not appeared until the past decade. For example, Meier, Leblanc, and Tan reported that the actor-critic network can correct the beam trajectory of the storage ring with fewer variables [10]. Ruichun *et al.* realized the online correction of electron orbits in the high-performance electron storage ring of Shanghai Synchrotron Radiation Facility by using the neural network trained by BPM historical data [11]. Similar work was also reported by the Beijing Electron Positron Collider [12]. Kain *et al.* reported that a sample-efficient RL algorithm normalized advantage function was successfully trained at the CERN AWAKE electron line and the H^- accelerator LINAC4 [13]. Because of the efficient convergence algorithm, the online time of orbit correction can be shortened to 20 min at the fastest.

For most of the above work, the RL agents were first tested in the simulation environments. Because of the difference between simulated and real accelerators, online training is still needed when applying the agents to a real accelerator. In order to realize real-time orbit correction, it is necessary to improve the learning efficiency of the algorithm as much as possible, so as to minimize the online training time of the agent. In this paper, we propose a new scheme that the RL agents can be used directly without further online training processes. The basic architecture of the agents is TD3 [14], which is based on the deep deterministic policy gradient [15] algorithm and is especially suitable for solving continuous control problems. In this paper, several improvements are employed to enable the TD3 algorithm to be used for fast orbit correction: The BPM trend is incorporated into the algorithm which enables the agent to learn an invariant relationship of the process of orbit correction. And the reward function is designed to encourage the agent to make decisions toward smaller BPM readings by observing the trends of it. The method enables agents to understand BPM trends rather than simply learning the relationship between BPM readings and correctors. Therefore, the agents themselves are embedded with sufficient robustness to overcome the difference between the simulated and the real environments. To verify the effectiveness of the improvements, we used two TD3-based agents with different network structures: One agent used a physical prior-knowledge-based network, while the other agent used the original TD3 network structure.

The agents were first trained in the simulated environment built by software TraceWin [16]. Then we deployed the agents directly on the medium energy beam transport (MEBT) section of China Accelerator Facility for Superheavy Elements (CAFe II) [17,18]. The experimental results show that the agents can correct the orbit in the MEBT section to within 1 mm. And all corrections are done within 30 s. Moreover, due to the reconstruction of the observation for the agent, when applied to different lattices of different particles, the agents are still valid. Compared with previous studies, our RL agents can skip the online training processes and data collection, which shows wide potential for improving the beam commissioning efficiency of accelerators.

The paper is organized as follows. In Sec. II, a brief introduction of RL and the orbit correction scheme, as well as the simulation and experimental environments, is described. The two agents and the introduction of reward function are given with necessary details. The result and discussion are presented in Sec. III. The agent was first trained and tested on the virtual accelerator. Then the experimental results on CAFe II are given. A conclusion is made in Sec. IV.

II. METHOD

Figure 1 is the overview of our procedure for the orbit correction task. The agent is first trained in a simulated environment; then, we applied the agent on the MEBT section of CAFe II without any further online training procedure. The main challenge of our scheme is to overcome the difference between the simulation and the real environment. We designed the agents according to the experience of accelerator commissioning engineers; they first observe the trend of BPMs and then determine the corrector magnet strength by the trend. The experience inspired us to design agents to complete this task in a similar way. Thus, we combined the BPM's current reading and its historical change as the input of the actor network in the TD3 algorithm rather than just the BPM readings. Meanwhile, the reward function also guides the agent to correct the trajectory in the right direction. Here, two different TD3-based network structures of actor are used to verify the effectiveness of incorporating the trend of BPM change in the observation space.

A. Reinforcement learning

Orbit correction for accelerators is a typical optimization problem. Its goal is to make BPM readings close to 0 by adjusting the strength of the correctors. Reinforcement learning is well suited to this problem. The paradigm of RL is shown in the left part in Fig. 1, which is mainly composed of an agent, environment, state of environment (observation for agent), action, and reward. The agent decides an action by observing the current state of the environment. After the agent executes the action, the environment will move to a new state, and the environment will give a reward (positive or negative reward) for the new state. This process can be described as a Markov decision process, which means that the state of time $t+1$ is determined only by the state of time t. A training process of RL can be defined as that an agent seeks an optimal strategy by interacting with the environment to maximize the expected cumulative reward.

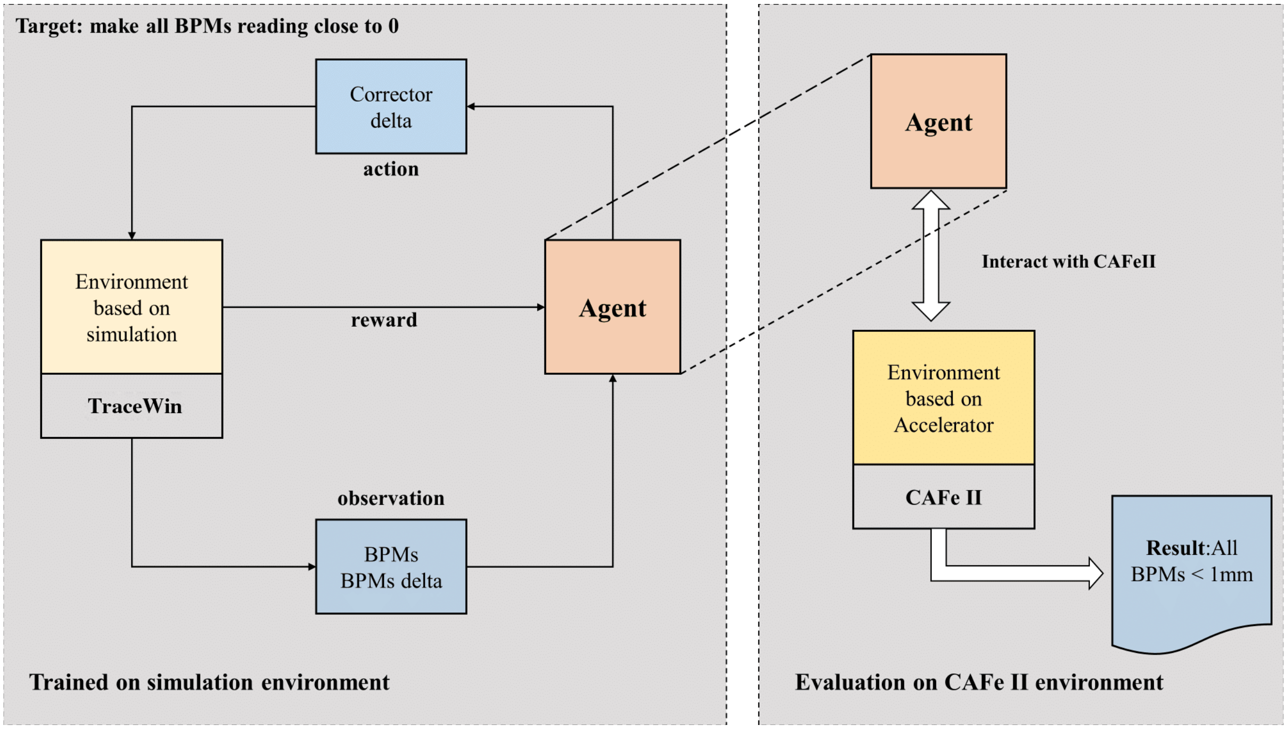
Target: make all BPMs reading close to 0
Corrector delta
action
Environment based on simulation
TraceWin
reward
Agent
observation
BPMs BPMs delta
Trained on simulation environment
Agent
Interact with CAFeII
Environment based on Accelerator
CAFe II
Result:All BPMs < 1mm
Evaluation on CAFe II environment

FIG. 1. Overview of the orbit correction method. The agent interacts with a simulation environment which is built by TraceWin. The agent determines the values of the corrector magnets according to the variation of the BPM readings. The goal of the agent is to maximize the expectation of reward in each episode. Once the model is trained on the simulated environment, it can be evaluated on CAFe II without any online data.

There are two main types of RL methods: value-based and policy-based methods. A value-based method learns a policy by maximizing the Q-value function, denoted $Q^{\pi}(s,a)$, which represents the long-term benefits of taking action a under state s by following policy π. It can be expressed as

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{T} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a\right], \qquad (1)$$

where $\mathbb{E}$ is the expectation value operator, T is the remaining steps from state $s = s_t$ till the end of the episode, and γ is the discount factor which indicates how much importance we want to give to future rewards. r is the reward, feedback from the environment when the agent performs action a in state s. A policy-based method learns an agent by seeking the optimal policy directly without the Q-value function. The actor-critic framework combines the value-based and policy-based into a more efficient method. The actor observes the state of environment and outputs the best action. The critic evaluates the action by calculating the Q-value function. With the training process of the policy network and value function network, the critic guides the actor to find the optimal policy by temporal difference error gradually, which is also one of the core ideas of TD3. An explore noise is added to the actions of the agent for exploring the environment adequately. In TD3, the actor is fixed for a certain number of steps while updating the two critics with each step. This improves the stability of the policy network in training process, and the number of fixed steps is defined as update frequency in this paper.

B. Environment

A virtual beam line is constructed by TraceWin according to the MEBT section of CAFe II. The structure of the MEBT section is shown in Fig. 2. The MEBT is a 2.6975-m long section consisting of six quadruple magnets (Q1–Q6) and two bunchers. Six groups of corrector magnets are set in the middle of quadruple magnets, while five groups of BPMs are installed.

The front five groups of corrector magnets are employed to correct the orbit of the beam in the x and y directions, while four groups of BPMs are used to probe the positions of the orbit. Thus, the input dimension of the actor is 8, and the output is 10. In order to simulate the offset of the beam, we added random errors for each element in TraceWin. With an error range of ±0.5 mm, the maximum deviation of the BPM readings can reach up to 9 mm. We develop a PYTHON wrapper to combine TraceWin and an environment to communicate with the agent based on OpenAI gym framework [19]. At the beginning of each episode, the currents of each corrector magnet are transmitted to TraceWin

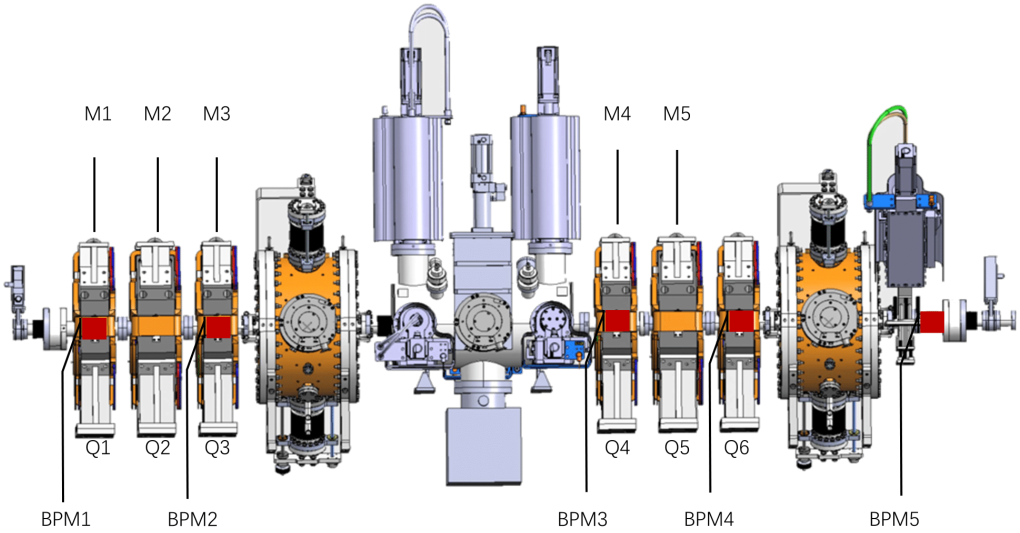

FIG. 2. The structure of the MEBT section of CAFe II. Q1–Q6 are the quadruple magnets, and M1–M5 are the correctors. Since the position of BPM1 is at the entrance of the MEBT section and the correctors cannot affect its reading, only BPM2–5 are used.

and dynamic simulations are performed. After simulations, each BPM reading is transmitted back to the agent. The agent should learn a policy to correct the orbit of the MEBT section by automatically selecting the current of all corrector magnets. The target of the agent is to make the rms of BPM readings less than 0.5 mm. The max time step of the orbit correction task is set to 30 as one episode in our experiments, which means that the agent should complete the mission in 30 time steps.

The CAFe II accelerator control system offers a PYTHON package (PYEPICS) to communicate with the hardware system. After training, we also built an environment on CAFe II, in which the corrector strength as well as the BPM readings are from the real accelerator instead of TraceWin.

C. Improvement of the agents based on TD3

The typical actor of TD3 is a neural network with an architecture as sketched in Fig. 3. The input of the network is BPM readings (B), followed by two full-connect hidden layers with 256 nodes. The output of the network is the strength of the correctors (M). However, this kind of input does not contain the trend of BPM reading evolution, of which is crucial in the orbit correction procedure.

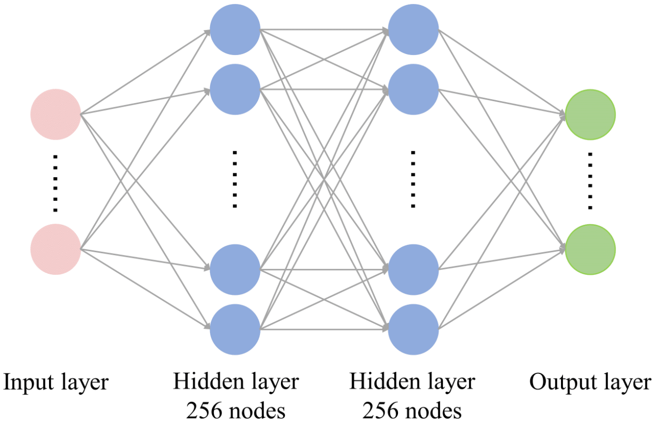

FIG. 3. Typical network structure of a TD3 actor.

In the actual accelerator, the installation positions of the BPM, the magnetic field of the correctors, and the beam state are different from those in the ideal simulation environment. If the BPM readings are directly taken as the observation in the TD3 network, the network trained in the simulated environment will not reflect the corrector-BPM mapping relationship in the real accelerator. In contrast, although there are differences between real and virtual accelerator environments, the main trend of corrector-BPM mapping is similar. Therefore, if the trend of BPM is taken as the observation of a TD3 network, agents trained in a simulated environment can also describe the corrector-BPM mapping relationship in a real accelerator to a certain extent, thus bridging the difference between virtual and real accelerator environments and enhancing the robustness of agents.

Thus, we propose a smart strategy in this paper: After changing the current of the corrector magnets (ΔM), a combination of both the BPM readings (B) and their changes (ΔB), rather than just the BPM readings alone, will be used as the observation value for the next step. We define the BPM reading at time $t = i$ as B_i and ${B_{xj}}^i$ as the reading of the jth x-direction BPM at time $t = i$. All eight BPM readings are defined as follows:

$$B_i = [{B_{x2}}^i, {B_{y2}}^i, {B_{x3}}^i, {B_{y3}}^i, {B_{x4}}^i, {B_{y4}}^i, {B_{x5}}^i, {B_{y5}}^i]. \quad (2)$$

The currents of the corrector magnets M_i at time $t = i$ are defined as follows:

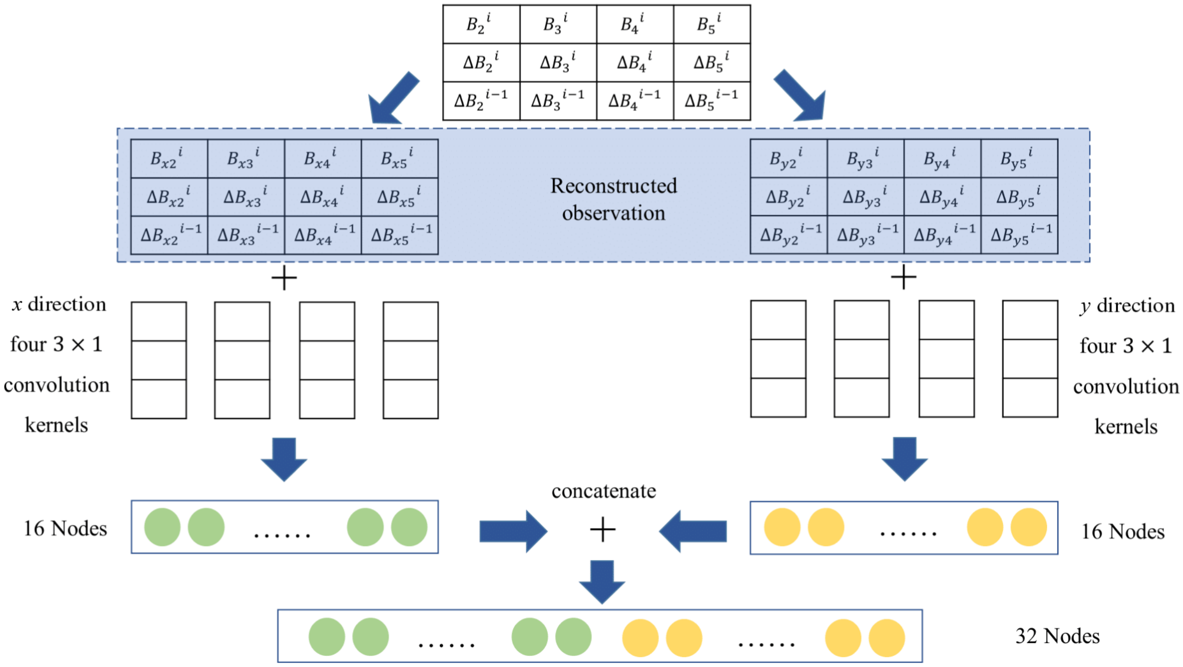
$B_2{}^i$ $B_3{}^i$ $B_4{}^i$ $B_5{}^i$
$\Delta B_2{}^i$ $\Delta B_3{}^i$ $\Delta B_4{}^i$ $\Delta B_5{}^i$
$\Delta B_2{}^{i-1}$ $\Delta B_3{}^{i-1}$ $\Delta B_4{}^{i-1}$ $\Delta B_5{}^{i-1}$
Reconstructed observation
x direction four 3 × 1 convolution kernels
y direction four 3 × 1 convolution kernels
16 Nodes
concatenate
16 Nodes
32 Nodes

FIG. 4. The convolutional layer between the input layer and the first hidden layer. With four 3 × 1 convolution kernels for both x and y directions, the output of each direction is a 16-dimensional vector; then join the two vectors directly as the input of the next hidden layer.

$$M_i = [M_{x1}{}^i, M_{y1}{}^i, M_{x2}{}^i, M_{y2}{}^i, M_{x3}{}^i, M_{y3}{}^i, M_{x4}{}^i, M_{y4}{}^i, M_{x5}{}^i, M_{y5}{}^i], \tag{3}$$

$$M_i = M_{i-1} + \Delta M_i, \tag{4}$$

where ΔM_i is the variation between time $t = i - 1$ and $t = i$ of the magnets.

To validate the effectiveness of incorporating the trend of BPM in the observation space, two actors with different network structure are designed to be tested. The first one is designed according to the experience of accelerator commissioning engineers. For simplicity, we call it the physical prior-knowledge-based network. To separately observe the variation trend of each BPM, we reconstructed the observation OB_{modified} as follows:

$$OB_{\text{modified}} = [\Delta B_{i-1}, \Delta B_i, B_i]^T, \tag{5}$$

$$\Delta B_i = B_i - B_{i-1}, \tag{6}$$

$$\Delta B_{i-1} = B_{i-1} - B_{i-2}, \tag{7}$$

where ΔB_i is the variation of BPM readings between $t = i$ and $t = i - 1$. Since the BPMs in the x and y direction in the MEBT section are decoupled, we separated the OB_{modified} in two parts, which is shown in the blue part in Fig. 4. Because of the properties of the observation, the agent needs to observe the state of BPMs column by column, 3 × 1 convolutional kernel is very suitable for observing the trend of each BPM in our case. Therefore, we added a convolutional layer between the input layer and the first hidden layer of TD3's actor network structure. Figure 4 illustrates the structure of the convolution layer. The second one used the original structure of the TD3 actor which is shown in Fig. 3, and the input of this actor OB_{original} is defined as follows:

$$OB_{\text{original}} = [\Delta B_{i-1}, \Delta B_i, B_i], \tag{8}$$

and the output is the variation of the correctors ΔM_i.

D. Reward function

The total reward r_{total} of a single time step is defined as follows:

$$r_{\text{total}} = r_{\text{distance}} + r_{\text{trend}} + r_{\text{value}}, \tag{9}$$

where r_{distance} is the distance reward, r_{trend} is the trend reward, and r_{value} is the value reward. The distance reward is defined as the Euclidean distance between the BPM readings and zero. Its formula is as follows:

$$r_{\text{distance}} = -\frac{1}{2N}\sqrt{\sum_{j=0}^{N}[(B_{xj})^2 + (B_{yj})^2]}, \tag{10}$$

where N is the total group of BPMs, in our case $N = 4$. The second component of the reward is referred to as the trend reward. The absolute difference of BPM readings at $t = i$ and $t = i - 1$ is calculated first, which is defined as $\Delta B_{abs} = |B^i| - |B^{i-1}|$. Then, we iterated through each element Δb in the vector ΔB_{abs}, and the reward for each element $r_{\text{trend\_single}}$ is defined as follows:

$$r_{\text{trend\_single}} = \begin{cases} 1/N, & \Delta b < 0, \\ -3/2N, & \Delta b \geq 0. \end{cases} \tag{11}$$

The total trend reward r_{trend} is the sum of all the $r_{\text{trend\_single}}$ of each element. The value reward r_{value} has been designed to make the BPM reading smaller during the tuning process. To calculate r_{value}, each element b of the vector B_i should be iterated, the reward $r_{\text{value\_single}}$ for each element is defined as follows:

$$r_{\text{value\_single}} = \begin{cases} -|b| \times 2, & |b| > 1, \\ (1 - |b|) \times 2, & |b| \leq 1, \end{cases} \tag{12}$$

and the r_{value} is the sum of all $r_{\text{value\_single}}$.

To minimize the number of steps required for the agent to complete the correction, the remaining steps in each episode are also used as part of the reward. When the agent corrects all of the BPM readings below 1 mm within 30 time steps, the agent will receive five points as the reward.

E. Hyperparameters

We tested the two types of agents in a simulated environment. One is based on the original network of TD3. The network is composed of four layers, including an input layer and two hidden layers with 256 nodes each and an output layer with ten nodes. The physical prior-knowledge-based one adds the special convolutional layer between the input layer and the first hidden layer. The learning rate was set to 3×10^{-4} for both actor and critic neural networks during the training process, and the batch size was set to 512. The explore noise of the agent was set as 0.2, and policy update frequency was set to 15, both of which are crucial for the TD3 agent. The range of ΔM was limited to −0.5 to 0.5. For the Q-value function in TD3, the discount factor of gamma was set to 0.98.

III. RESULT AND DISCUSSION

A. Simulation

At the beginning of each episode, we set the initial current values as M_0 and M_1 for the correctors:

$$M_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], \tag{13}$$

$$M_1 = [0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3]. \tag{14}$$

All of the missions start from this initialized status for the agent. The two agents are first trained in the simulated environment with the lattice of $^{40}\text{Ca}^{13+}$, whose parameters are shown in Table I and the result is shown in Fig. 5. According to the results shown in Fig. 5, the trained model at 15 000 time steps was chosen to evaluate on the simulated environment. The results are shown in Fig. 6.

TABLE I. Quadruple magnetic strength of $^{40}\text{Ca}^{13+}$, $^{55}\text{Mn}^{18+}$, and a proton in the MEBT section. The values are the magnetic field gradient of quadruple magnets, which will be setting to both TraceWin environment and CAFe II.

Particles	Q1 (T/m)	Q2 (T/m)	Q3 (T/m)	Q4 (T/m)	Q5 (T/m)	Q6 (T/m)
$^{40}\text{Ca}^{13+}$	15.80	−19.35	14.95	−11.95	14.55	−9.41
$^{55}\text{Mn}^{18+}$	13.50	−18.32	14.80	−11.28	13.46	−8.22
Proton	4.33	−5.72	4.81	−3.99	4.62	−2.53

Figure 6(a) shows the results of applying the agents trained by $^{40}\text{Ca}^{13+}$ to the orbit correction of $^{40}\text{Ca}^{13+}$. Figure 6(a)(1) shows how many steps the agents spent to correct the orbit rms of $^{40}\text{Ca}^{13+}$ to less than 0.5 mm in each of the 20 experiments. Figure 6(a)(2) presents a comparison between the initial orbit rms and the corrected orbit rms for the two agents. Figures 6(a)(3) and 6(a)(4) are one random sample in the 20 sets of experiments which show the orbit before and after correction in the x and y directions, respectively. The results indicate that the agent based on physical prior knowledge can correct the rms of the orbit to below 0.5 mm within 18 time steps, while the agent based on the original TD3 network structure can achieve this in six time steps.

In order to verify the robustness of the model, we added a ±30% random error to the strength of all quadrupole magnets, which is a large-scale error for the lattice. The results are shown in Fig. 6(b). Both of the agents correct the orbit to better than the 0.5 mm rms target within nine time steps. The result shows that our agents are not sensitive to

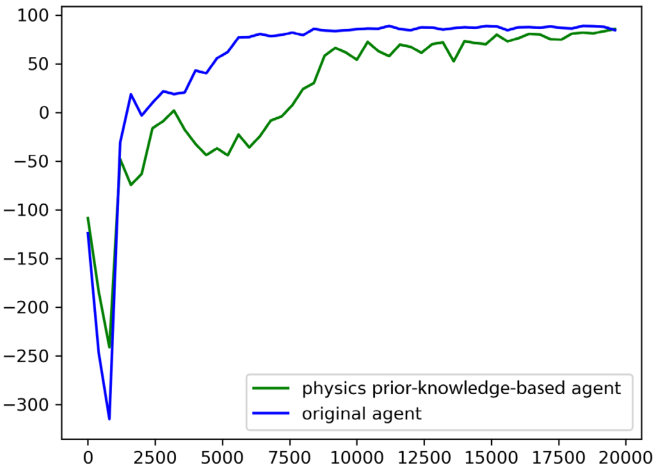

FIG. 5. The smoothed reward curves during the training process.

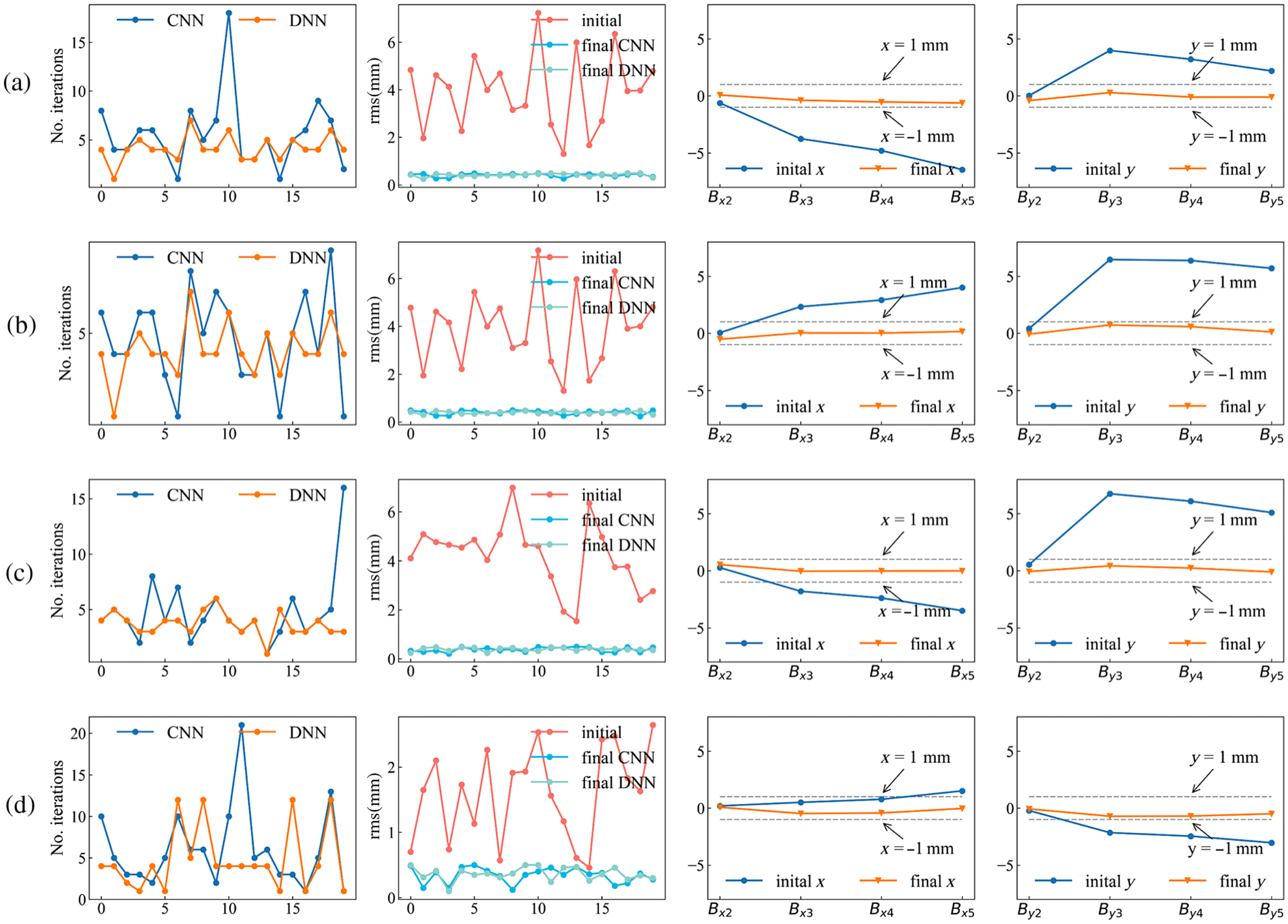

FIG. 6. 20 sets of experiments were performed for the two agents, using the same rms value as the initial setting for each experiment in every set. In this chart, CNN represents the agent based on physical prior knowledge, and DNN represents the agent with the original network structure of TD3. (a) Use the agents trained with ${}^{40}\mathrm{Ca}^{13+}$ to correct the orbit of itself. (b) Use the agents trained with ${}^{40}\mathrm{Ca}^{13+}$ to correct the orbit of itself with additional ±30% random error in the origin lattice. (c) Use the agents trained with ${}^{40}\mathrm{Ca}^{13+}$ to correct the orbit of ${}^{55}\mathrm{Mn}^{18+}$. (d) Use the agents trained with ${}^{40}\mathrm{Ca}^{13+}$ to correct the orbit of the proton.

the error of the quadruple magnet strength; they also can correct the orbit effectively in this situation. The incorporation of the BPM trend mentioned in Sec. II can effectively improve the robustness of the agents, which enables the agent to complete the orbit correction task when there is a large quadruple magnet error in the lattice. This is extremely crucial for the agents to switch from a simulated environment to a real accelerator environment directly.

Because of the strong expansibility of the algorithm, these agents can not only solve the difference between a real and a virtual accelerator, but also can be used to correct the orbit of different lattices. Moreover, for particles with different specific ratios, the only difference in their transmission in the MEBT section is that they have different responses to the magnetic field, which is similar to the case of particles transmitting in different lattices. Therefore, these agents have potential to be applied to lattices of different particles. We attempted to use the agent trained with ${}^{40}\mathrm{Ca}^{13+}$ to correct the orbit of ${}^{55}\mathrm{Mn}^{18+}$ and a proton. Parts of lattice parameters are shown in Table I. These findings in Figs. 6(c) and 6(d) are consistent with the concept that the agent is not sensitive to the type of particle. Both of the agents correct the orbit of ${}^{55}\mathrm{Mn}^{18+}$ better than 0.5 mm rms within 16 time steps and a proton within 21 time steps.

B. Online correction on CAFe II

The physical prior-knowledge-based agents trained in simulated environments are applied on CAFe II without any further operation. Three experiments are performed: (i) agent trained in a simulated ${}^{40}\mathrm{Ca}^{13+}$ lattice to the real ${}^{40}\mathrm{Ca}^{13+}$ lattice, (ii) agent trained in a simulated ${}^{55}\mathrm{Mn}^{18+}$ lattice to the real ${}^{40}\mathrm{Ca}^{13+}$ lattice, and (iii) agent trained in a simulated Mn lattice to the real proton lattice. The parameters of different lattices are shown in Table I and

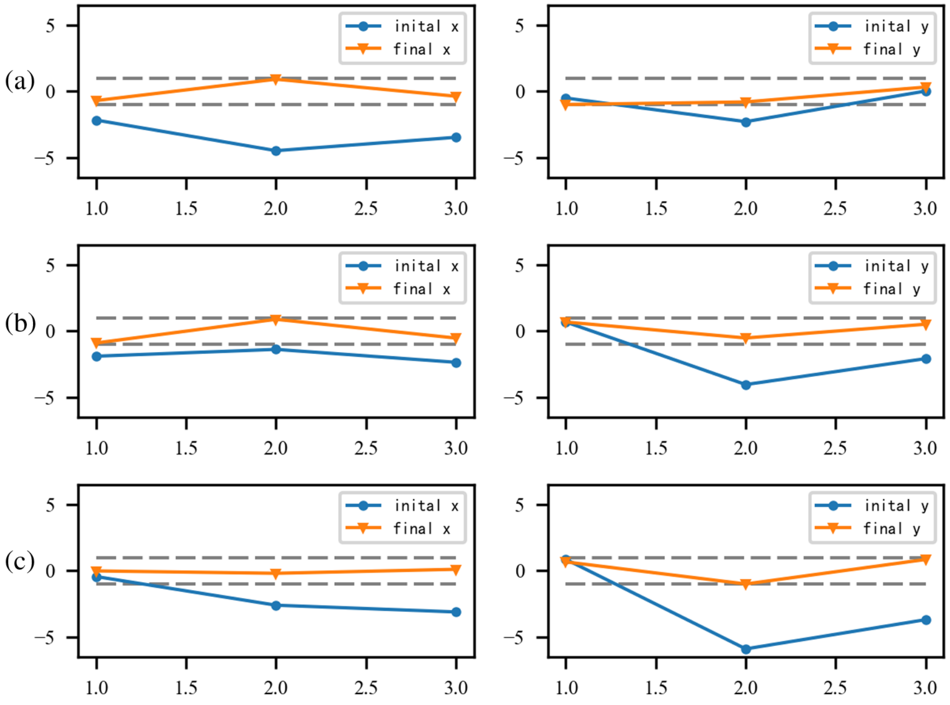

FIG. 7. The results of the agent on different beam lines. (a) Trained with $^{40}Ca^{13+}$ in a simulation environment and applied to $^{40}Ca^{13+}$ on CAFe II. (b) Trained with $^{55}Mn^{18+}$ in a simulation environment and applied to $^{40}Ca^{13+}$ on CAFe II. (c) Trained with $^{55}Mn^{18+}$ in a simulation environment and applied to a proton on CAFe II. The gray dotted line represents that the BPM reading is equal to 1 or −1 mm.

the correction results in Figs. 7(a)–7(c), respectively. The fifth BPM in CAFe II broke down during the experiments, so we set its parameters to zero in all cases and three BPMs are used as the observation actually. Figure 7(a) shows the result of using the agent trained with $^{40}Ca^{13+}$ in a simulation environment and applied to $^{40}Ca^{13+}$ on CAFe II. The agent corrected the orbit of $^{40}Ca^{13+}$ better than 1 mm within six time steps and 13 s, and the final rms is 0.72. The agent trained with $^{55}Mn^{18+}$ corrected the orbit of $^{40}Ca^{13+}$ better than 1 mm within five time steps and 11 s, and the final rms is 0.64, which is illustrated in Fig. 7(b). Figure 7(c) presents the result of using the agent trained with $^{55}Mn^{18+}$ and applied to a proton on CAFe II. The agent corrected the orbit of the proton better than 1 mm within 14 time steps and 30 s, and the final rms is 0.6, which is in good agreement with the results of the simulation experiment. The model we trained in a simulation environment can complete the task of orbit correction on a real accelerator without any retraining. And the agent is not sensitive to the error of quadruple magnets. The agent was deployed on an industrial personal computer for CAFe II control without a GPU, and all orbit correct tasks have been completed within 30 s in the real accelerator, which enables the agent to be used in a fast orbit correction task. The experiments on CAFe II have proved that the agent we trained on a simulation environment can apply to the real accelerator without any further data collection or retraining. Our agent is efficient enough for CAFe II; even the lattices of simulation and real accelerators are quite different.

IV. CONCLUSION

For fast orbit correction, the main challenge is how to overcome the gap between real and simulated environments, so that the agents trained in the simulations can be directly applied to real accelerators. This paper proposed a novel approach to enhance the reinforcement learning method for achieving rapid orbit correction by incorporating the BPM trend. TD3 algorithm is adopted as the basic architecture of the agents, while several improvements are made to realize robustness. These improvements enable the agent to understand the evolution trend of the BPM in the orbit correction process rather than simply learning the relationship between BPM readings and magnetic strength of correctors. Once the training is completed, the agents will adjust the beam trajectory based on the BPM evolution. Thus, the agents are not highly dependent on lattice parameters or particle types. The effectiveness of both the physical prior-knowledge-based agent and the original TD3 agent was verified in the simulated environment, and the results indicate that they both have strong robustness.

We applied the physical prior-knowledge-based agent to the MEBT section of CAFe II, and the results indicate that the agent can correct the orbit of $^{40}Ca^{13+}$ to better than 1 mm within 15 s. Considering the powerful generalization ability of the agent, we applied the agents to different lattices of different particles on CAFe II. The agent is first trained on a $^{55}\mathrm{Mn}^{18+}$ lattice in a simulated environment, and then it is used to correct the $^{40}\mathrm{Ca}^{13+}$ and proton lattices on CAFe II. Both of the orbits can be corrected to within 1 mm, and the time consumed is 11 and 30 s, respectively. The results are consistent with the concept that our agent is not sensitive to the lattice. These RL agents will have wide potential for improving the beam commissioning efficiency of accelerators. The approach we proposed to RL may find promising applications in other areas of accelerator operation.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. U22A20261) and Large Research Infrastructures “China initiative Accelerator Driven System” (Grant No. 2017-000052-75-01-000590).

[1] R. Nagaoka, C. J. Bocchetta, F. Iazzourene, E. Karantzoulis, M. Plesko, L. Tosi, R. P. Walker, A. Wrulich, and S. Trieste, Orbit correction in ELETTRA, *Proceedings of 4th EPAC* (1994), p. 1009.

[2] H. J. Tsai, H. P. Chang, P. J. Chou, C. C. Kuo, G. H. Luo, and M. H. Wang, Closed orbit correction of TPS storage ring, *Proceedings of 10th EPAC* (2006), p. 2029.

[3] J. Li, G. Liu, W. Li, B. Sun, C. Diao, and Z. Liu, Closed orbit correction of HLS storage ring, *19th IEEE Particle Accelerator Conference* (2001), p. 1255.

[4] J. Lygeros, D. N. Godbole, and S. Sastry, A design framework for hierarchical, hybrid control, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings, 1997, https://ideas.repec.org/p/cdl/itsrrp/qt1bk267px.html.

[5] N. Bruchon, G. Fenu, G. Gaio, M. Lonza, F. A. Pellegrino, and E. Salvato, Toward the application of reinforcement learning to the intensity control of a seeded free-electron laser, in *Proceedings of the 2019 23rd International Conference on Mechatronics Technology (ICMT), Salerno, Italy* (2019), pp. 1–6, 10.1109/ICMECT.2019.8932150.

[6] N. Bruchon, G. Fenu, G. Gaio, M. Lonza , F. H. O’Shea, F. A. Pellegrino, and E. Salvato, Basic reinforcement learning techniques to control the intensity of a seeded free-electron laser, Electronics **9**, 781 (2020).

[7] T. Boltz, M. Brosi, E. Bründermann, B. Hrer, and A. Müller, Feedback design for control of the micro-bunching instability based on reinforcement learning, in *Proceedings of the ICFA mini-Workshop on Mitigation of Coherent Beam Instabilities in Particle Accelerators* (2020), pp. 227–229, 10.23732/CYRCP-2020-009.227.

[8] A. L. Edelen, S. V. Milton, S. G. Biedron, J. P. Edelen, and P. J. Slot, Using a neural network control policy for rapid switching between beam parameters in an FEL, in *Proceedings of the 38th International Free Electron Laser Conference* (2017), pp. 406–409, 10.18429/JACoW-FEL2017-WEP031.

[9] E. Bozoki and A. Friedman, Neural networks and orbit control in accelerators, *4th European Particle Accelerator Conference* (1994), p. 1589.

[10] E. Meier, G. Leblanc, and Y. E. Tan, Orbit correction studies using neural networks, *Proceedings of IPAC2012* (2012).

[11] L. Ruichun, Z. Qinglei, M. Qingru, J. Bocheng, and Z. ZhenTang, Application of machine learning in orbital correction of storage ring, High Power Laser Part. Beams **33**, 034007 (2021).

[12] C. Z. Xiao Dengjie and Qiao Yusi, Orbit correction based on machine learning, High Power Laser Part. Beams **33**, 054004 (2021).

[13] V. Kain, S. Hirlander, B. Goddard, F. M. Velotti, and G. Valentino, Sample-efficient reinforcement learning for CERN accelerator control, Phys. Rev. Accel. Beams **23**, 124801 (2020).

[14] S. Fujimoto, H. van Hoof, and D. Meger, Addressing function approximation error in actor-critic methods, arXiv:1802.09477.

[15] 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, arXiv:1509.02971.

[16] D. Uriot and N. Pichoff, TraceWin manual, https://www.dacm-logiciels.fr/tracewin.

[17] Y. He, Stability and reliability study on the China ADS Front end superconducting demo linac (CAFe) (2019).

[18] Y. He, Q. Chen, W. Chen, Y. Chen, W. Dou, C. Feng, Z. Gao, G. Huang, H. Jia, T. Jiang, S. Liu, Z. Wang, F. Yang, S. Zhang, and H. Zhao, CiADS project: Next phase and linac commissioning results (2022).

[19] http://gym.openai.com.