ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA099

REINFORCEMENT CONTROL AND MATCHING FOR LEBT AND RFQ OF LINEAR ACCELERATORS*

Chunguang Su[1,2,3†], Zhijun Wang[‡1,2,3], Xiaolong Chen[1,3], Yongzhi Jia[1,2,3], Xin Qi[1,2,3], Duanyang Jia[1,2,3]
[1]Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Advanced Energy Science and Technology Guangdong Laboratory, Huizhou, China

Abstract

The reinforcement learning(RL) algorithm is applied to control the accelerator, with the aim of improving the transmission efficiency of the radio frequency quadrupole (RFQ) to achieve high beam intensity, reducing the debugging time, and improving the operation efficiency of the accelerator. To obtain high beam intensity, the RFQ transmission efficiency is crucial. A neural network model is established to partially replace the Tracewin software used for RL training. The soft actor-critic (SAC) algorithm trained on the neural network model successfully achieves RFQ transmission efficiency above 95% under different circumstances by controlling the low-energy beam transport (LEBT) solenoids. The accelerator control based on reinforcement learning has good generalization ability to cope with changes in different circumstances, and so has great potential in accelerator control.

INTRODUCTION

The light particle injector is a prototype accelerator of Light Particle Injection Platform for high-current particle injection, it can effectively accelerate proton beams of 10 mA and 1.5 MeV continuous wave, as well as milliamp-level helium beams of 6 MeV continuous wave. As shown in the Figure 1, The Light Particle Injection Platform consists of LEBT, RFQ, the beam control system, and the experimental target station, is an ideal research platform for semiconductor irradiation on a scale of several tens of micrometers. For this platform, achieving high beam currents is a crucial goal, and one of the key factors for achieving it is ensuring good configuration of the LEBT to improve the transmission efficiency of the RFQ. Recently, machine learning(ML) algorithms are widely used in accelerator beam commissioning areas [1, 2], such as machine learning-based beam orbit control [3] and longitudinal control of electron beams [4].

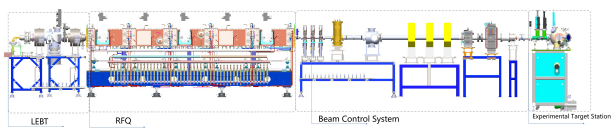

Figure 1: Layout of the Light Particle Injection Platform.

* Work supported by large research infrastructures China initiative accelerator driven system (2017- 000052-75-01-000590)
† Email: suchunguang@impcas.ac.cn
‡ Corresponding author, Email: wangzj@impcas.ac.cn

The desired target of RL to control LEBT elements is to achieve RFQ transmission efficiency above 95% constrained by LEBT efficiency above 75%. Results show method can effectively meet target without repetitive operation like manual adjustment or scanning and retraining strategies for different intensities or twiss parameters. Ultimate aim is verifying RL-trained agents' feasibility in optimizing light particle injector efficiency via simulations. During RL training, adjusting hyperparameters requires thousands of interactions. Initially, TraceWin was used but 4 minutes per interaction made adjusting one hyperparameter take about 40000 minutes (assuming 10000 interactions needed after each adjustment), which is unacceptable.

To address this, a NN model was built using Tracewin-generated data, replacing it as the interactive training environment, as illustrated in Figure 2. With each hyperparameter adjustment, training time was reduced to tens of minutes, substantially cutting RL training duration. Validation done on Tracewin, testing generality by varying beam intensity/twiss parameters and adding space charge effects. Next step involves experimental verification and application on real Light Particle Injection Platform for operational efficiency improvement.

METHOD

Reinforcement Learning

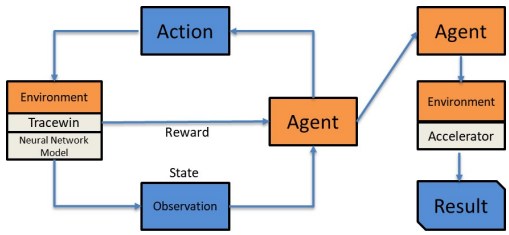

Figure 2: Training process and future plan.

The transmission efficiency regulation of RFQ is a typical optimization problem, where the goal is to achieve high transmission efficiency of RFQ by adjusting the current intensity of LEBT solenoids. Reinforcement learning is applicable for solving this problem. As illustrated in the Figure 2, RL is composed of five essential components, namely the agent, environment, state, action, and reward. The agent interacts with the environment through actions, receives feedback, and accumulates rewards over time through trial and error, with the objective of maximizing the expected cumulative

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA099

reward in the future and improving its action policy. The present work utilizes the maximum entropy reinforcement learning algorithm - SAC algorithm, which belongs to the actor-critic framework [5] [6] and is a reliable and efficient approach.

Relational Mapping Substitution Based on NN

The operation of beamline LEBT and RFQ were simulated with Tracewin software on a server with an eight-core CPU, with an initial particle count of 10,000. However, each simulation process took about 4.5 minutes, which was too slow for RL training and could not meet the training requirements of RL. Therefore, we collected 10,000 sets of data generated by Tracewin and used a simple NN to implement an alternative mapping of the relationship between the current values of the two LEBT solenoids and the beam intensity at the exit of LEBT and RFQ.

The model takes the current values of two solenoids as input and outputs the beam intensity at the exit of LEBT and RFQ. It comprises an input layer, an output layer, and two hidden layers, each with 16 nodes.

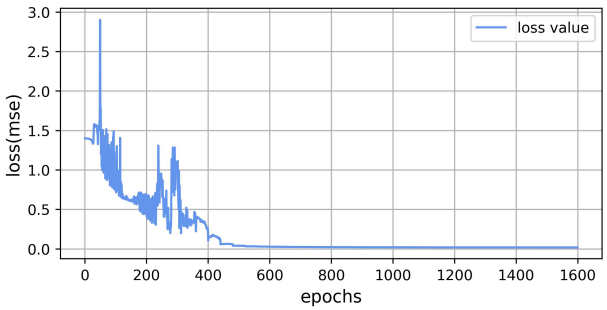

Figure 3: Training process of the neural network.

In this study, 10,000 Tracewin-generated datasets were used, allocating 8,000 for training and 2,000 for testing. During training process, as shown in Figure 3, mean squared error (MSE) was used as the loss function for neural network regression task evaluation. MSE is computed by squaring difference between predicted and true values, averaging it. Aim is to minimize sum of squared errors to make model predictions closer to actual values. Loss converged to 0.0182 after 1600 training sessions. During testing, NN model's coefficient of determination was 0.9871, a performance indicator used for evaluating fitting performance of regression models in neural networks, ranging between 0-1. indicates proportion of variance in dependent variable accounted for by independent variables. The closer it is to 1, the better the fit of the model.

Environment and Training

Two interactive environments were constructed to interact with agent: the first based on TraceWin, used for RL policy training and NN model reliability testing. To integrate it with environment that can communicate with agents, a Python wrapper was developed using OpenAI Gym [7]. However, due to lengthy Tracewin-based training process, an NN model was created as training environment and interacted via function calls in Python. Hyperparameters/reward functions adjusted multiple times to train appropriate reinforcement learning strategy.

During policy training, RL agent trained from same initial state: 5mA beam current delivered to LEBT entrance, solenoid coils initialized with SOL_CUR1 and SOL_CUR2 both set to 180.5A. Currents range from 133A to 228A, determined by Light Particle Injection Platform's safety range engineering design value. Policy training divided into exploration and evaluation stages. Exploration stage involves agent randomly trying different actions to discover rewards and states, while evaluation stage entails agent using learned strategy to perform tasks and computing average cumulative reward as performance metric. These two stages alternate and gradually learn optimal strategy. Exploration stage set to 5000 time steps, evaluation occurred every 1000 steps. Maximum time step for desired target search set to 50, meaning agent will stop at the 50 time step or the step which completed task.

Reward Function and Hyper-parameters

The reward function maps the states and actions of an agent to a scalar value, quantifying the desirability of the selected action towards achieving task objectives. By receiving reward signals from the environment, the agent adjusts and learns how to maximize the long-term cumulative discounted rewards in order to achieve the desired target. In this paper, the setting of reward function was listed in the following formula:

$$Reward_{delta} = \begin{cases} +0.1, delta > 0 \\ -0.3, delta < 0 \end{cases} \quad (1)$$

$$Reward_{LEBT} = \begin{cases} +0.05, rfq\_in > 3 \\ -1, rfq\_in < 3 \end{cases} \quad (2)$$

$$Reward_{RFQ} = \begin{cases} (1 - Trans) \times 3, Trans > 0.9 \\ (Trans - 0.9) \times 2, Trans < 0.9 \end{cases} \quad (3)$$

In equation (1), $delta$ represents the increase in RFQ transmission efficiency from the previous to the current time step. Based on the value of $Reward_{delta}$, if the current transmission efficiency is higher than the previous step, the agent is rewarded, otherwise it is punished. Equation (1) is employed to guide the agent to gradually improve the RFQ transmission efficiency. In equation (2), the $Reward_{LEBT}$ reflects the agent's performance in optimizing the beam current at the exit of the LEBT system. If the beam current exceeds 3 mA, the agent receives a reward, otherwise it is penalized. This reward function is adopted to prevent excessive beam losses in the LEBT system.

Q-value function in RL evaluates long-term expected return for taking specific action in current state with discount factor gamma and learning rate controlling magnitude of changes in network weights. Reward at each future time step

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA099

multiplied by power of gamma, which gradually decreases over time steps, allows agent to consider possible future rewards while focusing more on immediate payoffs. During agent training, hyperparameters set as follows: actor and critic's neural networks have learning rate of 3×10^{-5}. For SAC algorithm's Q-value function, gamma was set to 0.98. A softly update factor of 0.005 used for smoothing the target network's update.

RESULT

By utilizing a neural network model as the interactive environment for training, we obtained a policy that meets the expected results. As shown in Figure 4, when the initial values of two solenoid coils are both set to be 180.5A, the expected goal can be achieved within only 22 steps. After being validated with Tracewin as the interactive environment, the goal can be accomplished within only 8 steps, indicating that the policy obtained by replacing Tracewin with the nn model as the interactive environment is reliable. Due to the fact that the training process starts from the same initial values, the policy is sensitive to the initial values and can only be solved within the range where the solenoid coil initially varies between 177.5A-185A.

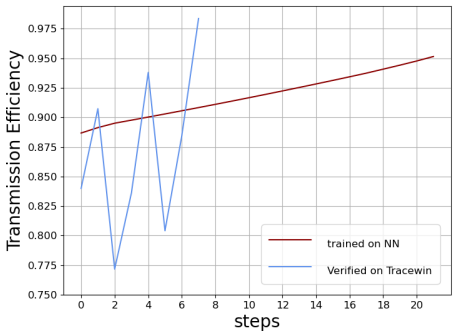

Figure 4: Testing process on NN and Tracewin.

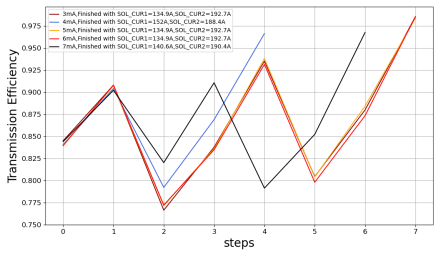

Figure 5: Test results of different beam intensities of LEBT entrance.

For Figure 5, it illustrates the performance of the policy under different entrance beam intensities in LEBT, and the results indicate that the policy can achieve the desired goal within a maximum of 8 time steps in this scenario. In Figure 6, Twiss1 corresponds to a beam with AlphaX and AlphaY

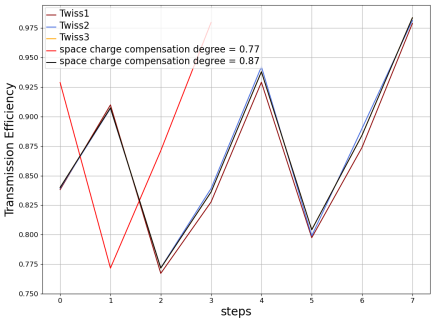

Figure 6: Test results of different beam twiss parameters of LEBT entrance and different space charge compensation degrees.

both being -1.6 mm/pi.mrad, BetaX and BetaY both being 0.144 mm/pi.mrad; Twiss2 corresponds to a beam with AlphaX and AlphaY both being -2 mm/pi.mrad, BetaX and BetaY both being 0.18 mm/pi.mrad; Twiss3 corresponds to a beam with AlphaX and AlphaY both being -2.4 mm/pi.mrad, BetaX and BetaY both being 0.216 mm/pi.mrad. The results show that the policy can achieve the desired target within a maximum of 9 time steps in this scenario. In addition, we tested the policy performance when the space charge compensation degree was at 87%, 77%, 67%, and 97%, as illustrated in Figure 6. The results indicate that the policy can achieve the desired target within a maximum of 8 time steps when the space charge compensation degree is at 87% and 77%. However, for the cases with a space charge compensation degree of 67% and 97%, it did not meet the desired target.

SUMMARY AND OUTLOOK

In this paper, Tracewin was replaced with NN model as interaction environment to train policy achieving desired targets, and model's control optimization ability was verified on Tracewin. Even under changed beam conditions like entrance beam intensity and twiss parameters, policy can still achieve desired target, demonstrating generality to some extent. However, policy couldn't meet desired target with space charge compensation degree of 67% and 97%, indicating further hyperparameter and reward function optimization, or more data for training may be necessary.

Next step is retraining policy to accomplish tasks under more complex conditions and achieving good performance through testing on Light Particle Injection Platform.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to the Large Research Infrastructures China initiative Accelerator Driven System Project (Grant No. 2017-000052-75-01-000590) for their generous support in the completion of this research work. Their financial assistance has been instrumental in enabling us to carry out this research successfully.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA099

REFERENCES

[1] A. Edelen *et al.*, "Opportunities in machine learning for particle accelerators," *arXiv preprint*, 2018. doi:10.48550/arXiv.1811.03172

[2] A. Scheinker, C. Emma, A. L. Edelen, and S. Gessner, "Advanced control methods for particle accelerators (ACM4PA) 2019," in *Proc. ACM4PA 2019*, 2019, pp. 1-24. doi:10.48550/arXiv.2001.05461

[3] Y. Hitaka *et al.*, "Numerical methods for the orbit control at the KEK 12 GEV-PS," in *Proc. EPAC'04*, Lucerne, Switzerland, Jul. 2004, paper THPLT073, pp. 2465-2467.

[4] A. Rezaeizadeh, T. Schilcher, and R. S. Smith, "Adaptive robust control of longitudinal and transverse electron beam profiles," *Phys. Rev. Accel. Beams*, vol. 19, no. 5, pp. 052802, Oct. 2016. doi:10.1103/PhysRevAccelBeams.19.052802

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the International Conference on Machine Learning*, 2018, vol. 80, pp. 1861-1870.

[6] T. Haarnoja *et al.*, "Soft Actor-Critic Algorithms and Applications," *arXiv preprint*, 2018. doi:10.48550/arXiv.1812.05905

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