ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA100

DEMONSTRATION OF BEAM EMITTANCE OPTIMIZATION USING REINFORCEMENT LEARNING

D. Marcato*,[1], L. Bellan, D. Bortolato, M. Comunian, A. Galatà, F. Gelain,
V. Martinelli, G. Savarese, INFN Legnaro National Laboratories, Legnaro, Italy
G. A. Susto, University of Padova, Department of Information Engineering, Padova, Italy
[1]also at University of Padova, Department of Information Engineering, Padova, Italy

Abstract

In Particle accelerators, commissioning of a complex beam line requires extensive use of computer models. When the as-built beam line cannot be exactly modeled by the simulation (due for example to mechanical errors or to the extensive usage of the non-linear focusing forces), the solution found in the simulations needs to be adjusted. Thus, it is often required to modify the settings by exploring different parameters ranges on the real accelerator. Given the high parameter space, this is a demanding task both in term of beam time and in term of required expertise. Furthermore, there is no guarantee to reach the optimal solution. This paper proposes a Reinforcement Learning approach to develop a model able to efficiently explore the parameter space of a beam line and iteratively move towards the optimal solution. The approach is first applied for the ADIGE Medium Resolution Mass Separator at INFN Legnaro National Laboratories, where the potentials of an electrostatic multipole must be correctly tuned to minimize the output beam emittance after the separation stage.

INTRODUCTION

Beam commissioning is a critical phase of the operation of a particle accelerators. It requires extensive use of computer models to simulate the real accelerator and its beam dynamics. The simulation applies some algorithms to optimize a certain number of beam properties, including the beam transmission and the beam emittance. The simulation results thus include the setpoint of all the beam transport and accelerating elements which correspond to the optimal beam parameters.

In practice, the task of perfectly modelling the real accelerator with a simulation it's not trivial. This task is especially complicated when the accelerator is old and there are many mechanical uncertainties or when the effect of non-linear focusing forces becomes relevant. In these cases the results from the simulation can't be directly used, because the optimal solution may be quite different from the one derived by the simulation leading the beam transport may fail completely. Thus, it's usually required to improve the original solution by manually exploring the parameter space of all the beam transport elements to achieve the best beam dynamic. This operation is a labor-intensive task, which requires high skills and experience. The more parameters are available, the harder it is to converge to a working solution, with no guarantee of reaching the optimal one.

Machine Learning (ML) and Deep Learning (DL) models are suitable candidates to solve this kind of tasks, characterized by a big parameter space and an output function to optimize. Furthermore, by learning from real-world data these methods could overcome the limitations of the simulations and reach a more accurate solution. For these reasons there is now great interest on this field towards ML/DL solutions [1]. For example, Ref. [2] presents multiple machine learning applications to beam dynamics problems at CERN, including a model for the beam commissioning procedure to setup the collimators. In Ref. [3], a Reinforcement Learning (RL) agent is trained to optimize the beam intensity of a linear accelerator. DL approaches have the condition of having a priori tagged data, which is typically unfeasible in many scenarios; RL approaches instead collect information and training data by directly interact with the environment.

This paper proposes a novel approach based on a RL model to tune online the control system parameters of a particle accelerator and obtain the minimal beam emittance. As a first demonstration of feasibility, the approach is tested on the ADIGE MRMS beam line at INFN Legnaro National Laboratories (LNL). This beam line includes an electrostatic multipole with 48 independent voltage terminals, and thus represents the perfect example of a beam transport element which is hard to tune manually.

ADIGE MRMS

The Medium Resolution Mass Separator (MRMS) [4] is installed on the ADIGE 1+ ion source beam line after a charge breeder, to separate the contaminants introduced by

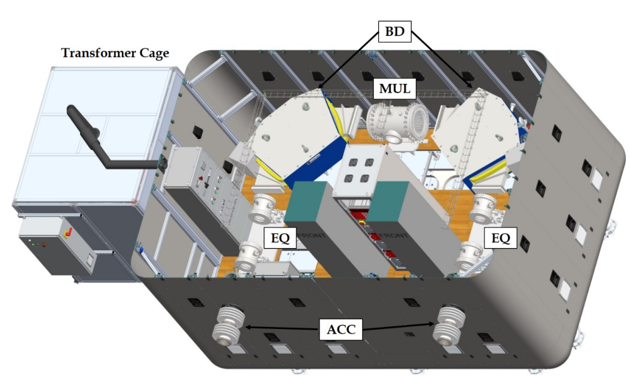

Figure 1: The MRMS platform [4].

* davide.marcato@lnl.infn.it

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA100

the breeding stage. It is composed of a high voltage platform operating at -150 kV with 4 electrostatic quadrupoles (up to 12 kV), 2 bending dipoles and an electrostatic multipole between them. This is a cylinder composed of 48 high voltage terminals operating at ±2.5 kV, so that it can be used as a high order multipole. Finally, on the beam line after the platform an Allison Scanner is available to measure the beam emittance.

At the entrance of the platform the beam is defocused on the horizontal axis before the dipoles to maximize their resolving power. This amplifies the non-linear effects to the beam passing through the dipoles, and increases the beam emittance. The multipole is designed to compensate these non-linear effects and maintain a good emittance value. By having a large number of high voltage terminal it can be easily reconfigured to act as a quadrupole, esapole, etc. or with a generic configuration up to 48 poles. Thus it is able to compensate higher order aberrations of the beam.

Finding the correct configuration of all the voltage values of the multipole is not trivial, given its high parameter space and its strong dependency on the details of the real machine. For this reasons, this problem is used as a first demonstration of the usage of RL models for beam commissioning. We want thus to develop a model which, given the real machine, is able to iteratively converge to the multipole configuration which minimizes the beam emittance after the MRMS. This model can be trained on a simulation but over time it can be fine-tuned on the data from the real machine. The model should be generic enough to be able to reach the correct solution even when the optimal value changes: our goal is for the model to not learn a single solution but to be able to iteratively move towards the optimal solution.

PROPOSED METHOD

In this Section, we present our approach for the development of a model able to find the optimal configuration of the multipole high voltages setpoint. The multipole has 48 independent terminals, and thus in theory our configuration should contain 48 parameters. In practice, we are not interested in generating a skewed field, and thus the terminals are connected to 24 power supplies in a symmetric way over the y axis. Furthermore, since the non-linear effects introduced by the dipoles follow the order of x^2, x^3, x^4, etc. where x is the size of the beam envelope on the x axis, we chose to correct them by using a linear combination of the fields of sextupole, octapole, decapole and dodecapole. This effectively reduces the number of parameters to 4 and forces the solution to follow the physics of the problem.

Then, with the following formula we can derive the voltage value on all the terminals (see Fig. 2):

$$\phi(\rho,\theta) = \sum_{n\in[3,4,5,6]} A_n \rho^n \cos(n\theta),$$

where ρ is the multipole cylinder radius, θ is the angle of each high voltage terminal, and A_n is the parameter to learn. This acts as a weight for each of the basic configurations

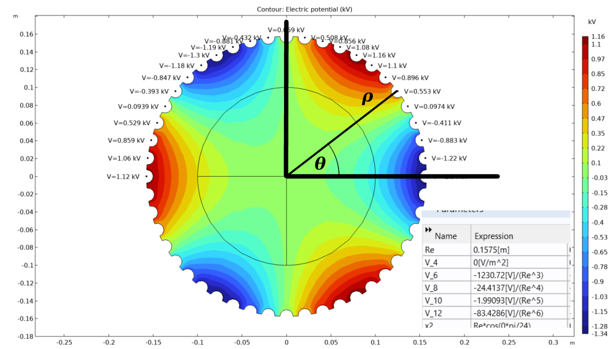

Figure 2: Example configuration of the multipole voltages.

(sextupole, octapole, decapole and dodecapole), so that the final solution is a linear combination of those.

Reinforcement Learning

Now we want to build a deep learning model to find the optimal A_n values. Since we don't want to learn a single solution, we build a RL model that learns to *move* the parameters towards the optimal solution. As can be see in Figure 3, the RL paradigm is composed of an agent, who receives an observation and decides to perform an action. As input observation we are using the x, x' beam emittance graph image, while the actions correspond to an adjustment of the 4 multipole parameters, which can be incremented or decremented. The new voltage setpoints are thus computed and a simulation with the new values is run, from which we get a new emittance graph and the emittance value.

During training this value is compared to the one from the previous simulation and, if the actions of the agent reduced the emittance, then a positive reward is given, otherwise a negative one. Hopefully, given enough episodes, the agent should learn how to converge towards the minimal beam emittance configuration. Finally, on the evaluation or deployment phase the model is used to change the multipole parameters according to its policy until it converges to a minimal solution.

A physics simulator, TraceWin, is used to simulate the beam dynamics of the line and obtain the beam emittance after the MRMS. A python wrapper has been developed to run the simulator from python code, enabling the integration of the simulator into the RL loop. To train the model the Stable-Baseline3 [5] library is used. This library implements

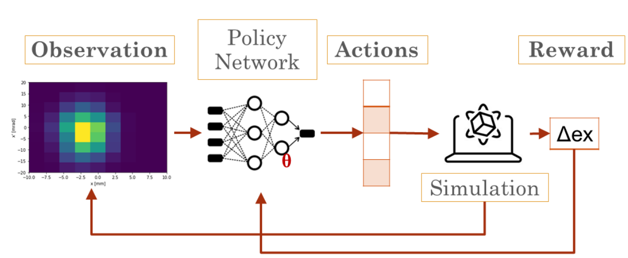

Figure 3: Reinforcement learning training and evaluation.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA100

several algorithms in PyTorch and exposes a unified user interface for all the algorithms. The library works natively on gym [6] environments but supports the creation of custom environments which should implement the standard interface of a gym environment. Thus an environment for a TraceWin simulation of the multipole was developed. Initially, when the environment is reset, the multipole parameters are set randomly with a uniform distribution in their operating range. Then, the TraceWin simulator is run and the results are recorded. The new state, that is the observation to be used as input of the agent, is computed as a 2D histogram of the particle distribution in x and x' on the diagnostic box after the MRMS. This returns normalized 36x36 vector, which can be considered an image, corresponding to phase space plot of the beam emittance. The resolution of such image is intentionally kept low to reduce noise and the size of the agent CNN network, thus reducing the training effort. Instead the reward of the n-th iteration is calculated with the following formula:

$$r^n = \left(\varepsilon_x^{n-1} - \varepsilon_x^n + \frac{I_l^{n-1} - I_l^n}{a} \right) * b,$$

where ε_x is the emittance value, I_l is the number of lost particles on the beam transport through the MRMS, a is a calibration factor which was set to $a = 10000$ (number of simulated particles) and $b = 10$ is a gain used to amplify the reward value. This rewards is higher the more the beam emittance is reduced compared to the previous iteration and the fewer particles are lost. This last term was introduces to avoid a solution where the beam emittance reaches its minimum due to heavy transmission losses. Finally, an episode is concluded when the emittance reaches a low threshold or when the number of iterations in the episode reach a limit of 10.

The environment is configured with a *Box* action space, that is a vector consisting of 4 real values between -1 and 1, which is then multiplied by a vector of coefficients ($[200, 100, 50, 50]$) to obtain the values to adjust the multipole setpoints. These coefficients were chosen by dividing the expected operating range of each parameter by the maximum number of iterations per episode.

As a first algorithm to demonstrate the feasibility of the method, the Proximal Policy Optimization (PPO) [7] algorithm was used due to its flexibility and popularity in the RL community. Since the input is a 2D image, the policy network was configured to use a Convolutional Neural Network (CNN) with an architecture which satisfies the constraints given by the input and output specifications.

RESULTS

The physics simulator is configured with a Sn19+ beam at 0.76 MeV. To correctly simulate the beam distribution the simulator calculates the complete dynamics of 10000 particles along the ADIGE beam line. This means that, even a high performance server, a single simulation takes about 15 seconds to complete. GPU acceleration is used for the neural network training, but the simulator does not support it and the simulation step is responsible for most of the computing time. This is obviously a big limiting factor on the amount of simulations that can be performed and thus on the complexity of our model.

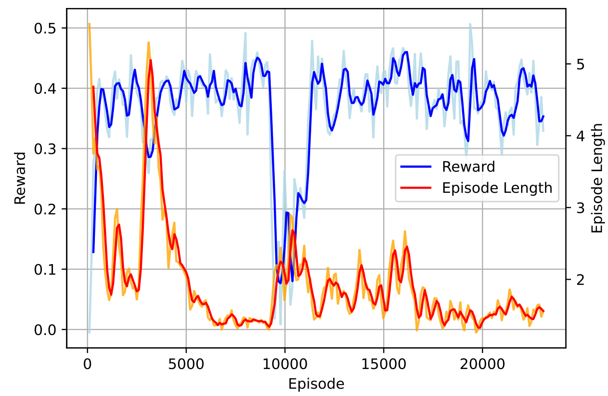

Figure 4: Mean reward and episode lenghts during training.

Given this setup, the agent is trained for a few days and about 25k episodes. As can be seen in Figure 4 the mean reward reaches a fairly constant positive value in relatively few episodes. Instead, the average episode length requires more episode to stabilize to a small value. This means that the training is actually successful and the model is able to reach the minimum beam emittance in few steps. Furthermore, by running the trained agent on new simulations we observed that the model is able to converge to a minimun 97% of the times with an average of just 2.2 steps.

CONCLUSIONS

In this paper, we presented novel a method based on reinforcement learning to train an agent which is able to iteratively optimize the parameters of an electrostatic multipole on the ADIGE beam line to minimize the beam emittance. This demonstrates the feasibility of using reinforcement learning models to automatically explore in a smart way the parameter space of a complex beam line and converge towards the optimal beam dynamics solution.

Nevertheless, the proposed method requires further research to scale it to more parameters and evaluate its performance with a greater environment variability. In particular, it would be interesting to test how much a trained model is able to adapt to a different input beam or slightly different beam line, thus generalizing to different use cases. Finally, data from the real machine could be used to further fine tune its behaviour and actually learn the peculiarities of a specific beam line.

REFERENCES

[1] D. Marcato *et al.*, "Machine learning-based anomaly detection for particle accelerators," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 240–246. doi:10.1109/CCTA48906.2021.9658806

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA100

[2] P. Arpaia *et al.*, "Machine learning for beam dynamics studies at the cern large hadron collider," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 985, p. 164 652, 2021. doi:10.1016/j.nima.2020.164652

[3] N. Madysa, R. Alemany-Fernández, N. Biancacci, B. Goddard, V. Kain, and F. M. Velotti, "Automated Intensity Optimisation Using Reinforcement Learning at LEIR," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 941–944. doi:10.18429/JACoW-IPAC2022-TUPOST040

[4] A. Galatà *et al.*, "Progresses in the installation of the spes-charge breeder beam line," *J. Instrum.*, vol. 13, no. 12, p. C12009, 2018. doi:10.1088/1748-0221/13/12/C12009

[5] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 1–8, 2021. http://jmlr.org/papers/v22/20-1364.html

[6] G. Brockman *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[7] J. Schulman *et al.*, *Proximal policy optimization algorithms*, 2017.