ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA098

INJECTION OPTIMIZATION VIA REINFORCEMENT LEARNING AT THE COOLER SYNCHROTRON COSY

A. Awal*,[1], J. Hetzel, GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany
J. Pretz[1], Institut für Kernpyhsik, Forschungszentrum Jülich, Jülich, Germany
[1]also at III. Physikalisches Institut B, RWTH Aachen University, Aachen, Germany

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Abstract

In an accelerator facility, it is crucial to have a particle beam with high intensity and small emittance in a timely manner. The main challenges restraining the availability of the beam to the user and limiting the beam intensity in storage rings are a lengthy optimization process, and the injection losses. The setup of the Injection Beam Line (IBL) depends on a large number of configurations in a complex, non-linear, and time-dependent way. Reinforcement Learning (RL) methods have shown great potential in optimizing various complex systems. However, unlike other optimization methods, RL agents are sample inefficient and have to be to be trained in simulation before running them on the real IBL. In this research, we train RL agents to learn the optimal injection strategy of the IBL for the Cooler Synchrotron (COSY) at Forschungszentrum Jülich. We address the challenge of sim-to-real transfer, where the RL agent trained in simulation does not perform well in the real world, by incorporating domain randomization. The goal is to increase the beam intensity inside COSY while decreasing the setup time required. This method has the potential to be applied in future accelerators like the FAIR facility.

THE COSY INJECTION BEAM LINE IBL

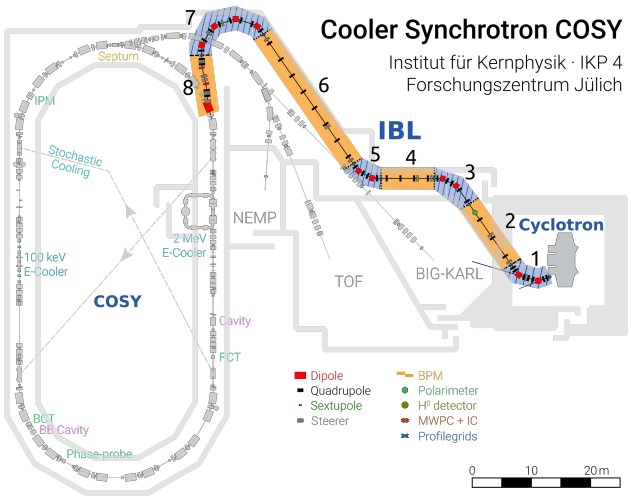

Figure 1: The COSY facility at the research centre in Jülich (FZJ), depicted are the cyclotron (right), the cooler synchrotron COSY (left), and the interconnecting injection beam line IBL. For the latter its division in sections is indicated with colors and numerals.

The topical injection beam line IBL is a single transfer beam line of the accelerator facility at the Research Centre in Jülich, Germany. Here preaccelerated negatively charged hydrogen- and deuteron ions from the cyclotron are transported to the Cooler Synchrotron COSY [1, 2], where they are accumulated and accelerated after being injected with multi-turn stripping foil (charge exchange) injection. An overview over the facility is given in Fig. 1. The beam line is 94 m long and the transported hydrogen (deuteron) ions have a kinetic energy of 45 MeV (76 MeV). The IBL can be subdivided into eight sections where bent sections alternate with straight sections. At the beginning of each section profile grids can be applied to measure the beam profile in horizontal and vertical directions. At the beginning of the sections 1, 2 and 8, a Faraday cup can be inserted to measure the intensity of the beam at these locations. Additionally, at the beginning and the end of the IBL a viewer screen can be inserted to record an image of the beam cross section.

Our efforts to optimise the whole IBL with a Bayesian Optimiser have been described recently [3]. In contrast in this work we focus on the optimisation of the injection with manipulation of section 8 only with Reinforcement Learning RL. Section 8 consists of 4 quadrupoles and 7 steerers that the reinforcement agent can manipulate autonomously. Its purpose is to match the transferred beam from the previous sections to the acceptance of COSY.

REINFORCEMENT LEARNING

A standard reinforcement learning (RL) problem involves an agent interacting with an environment by following a policy to maximize a reward. The state of the environment at each time step is denoted by $s_t \in S$. For simplicity, we assume that the state is fully observable. The policy $\pi(a|s)$ defines a probability distribution over actions given a state, where each query to the policy samples an action $a \in A$ from the conditional distribution. The reward function $r : S \times A \to \mathbb{R}$ provides a scalar value that reflects the desirability of performing an action at a given state. For convenience, we denote $r_t = r(s_t, a_t)$. Figure 2 shows the standard RL agent learning cycle. The goal of the agent is to find an optimal policy that maximizes the expected return over a horizon. The expected return is the sum of discounted rewards obtained during an episode starting from a fixed initial state. The multi-step return is given by:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$$

where $\gamma \in [0, 1]$ is a discount factor and T is the horizon of each episode. The state value function is the expected return

* a.awal@gsi.de

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA098

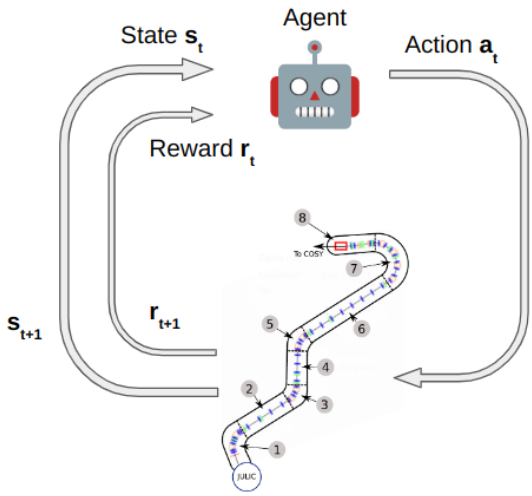

Figure 2: At each time step, the agent observes the current state s_t of the environment, selects an action a_t based on its current policy π, receives a reward signal r_{t+1} from the environment, and transitions to the next state s_{t+1}.

over the horizon $V(s_t) = \mathbb{E}[R_t|S = s_t]$. If each episode starts in a fixed initial state, the expected return of following a given policy can be rewritten as the expected return starting at the first step:

$$J(\pi) = \mathbb{E}[R_0|\pi] = \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right]$$

where $p(\tau|\pi)$ represents the likelihood of a trajectory $\tau = (s_0, a_0, s_1, ..., a_{T-1}, s_T)$ under the policy π, and the state transition model $p(s_{t+1}|s_t, a_t)$ is determined by the dynamics of the environment [4]. The objective during learning is to find an optimal policy π^* that maximizes the expected return of the agent $J(\pi)$:

$$\pi^* = \arg\max_\pi J(\pi)$$

In the domain of accelerators and many other real-world applications, it is difficult or expensive to obtain a large and diverse dataset of real-world examples to train an agent. In such cases, domain randomization in simulation can be used to generate a large number of simulated environments with different variations of the task, and use these environments to train the agent. This technique can help the agent to learn to adapt to different situations and generalize well to new, unseen environments. Additionally, domain randomization can also help to overcome the issue of overfitting, where an agent may memorize specific details of the training environments rather than learning to generalize to new environments [5].

REINFORCEMENT LEARNING AT COSY

The current injection optimization process at COSY is performed manually by adjusting the last section of the IBL, specifically section 8, which consists of 4 quadrupoles and 7 steerers. This optimization is carried out directly against the beam current inside COSY. However, this approach is time-consuming and does not guarantee consistent beam characteristics inside the storage ring.

An alternative approach involves optimizing the phase space of the beam at the injection point, rather than the beam current inside the storage ring. To achieve a high-intensity beam, the phase space of the injected beam should intersect with the acceptance of the storage ring and the strip foil. By ensuring a consistent phase space at the injection point, the beam characteristics inside COSY remains consistent. To implement this approach, a camera is deployed at the end of the IBL, and optimization is performed directly towards the camera outputs. The operator sets the target of the beam as μ_x, μ_y and σ_x, σ_y. $\mu_{x,y}, \sigma_{x,y}$ correspond to the center and focus of the desired beam respectively. The optimization is performed consequently to match the beam at the camera with the desired beam. By setting the right parameters, a consistent beam with high intensity inside the storage ring can be obtained.

To incorporate this process, we consider the reward to depend not only on the state and action but also on a stochastic goal g. The reward function subsequently becomes $r(s, a, g)$ and the agent's policy is then modified to incorporate the goal, resulting in $\pi(a|s, g)$ [6]. This allows the agent to learn a policy that is adaptable to various goals at the injection point.

Domain Randomization Training a policy under a single dynamics model may limit its performance when applied to real-world scenarios with different dynamics. To address this, we incorporate a range of dynamics variations in the training process, allowing the policy to adapt and perform well under different conditions.

These variations in the environment dynamics are managed using a set of parameters, denoted by ρ. At the beginning of each training episode, ρ is randomly sampled and remains constant throughout the episode. The transition dynamics of the environment are then defined as $P(s_{t+1}|s_t, a_t, \rho)$. By training the policy to adapt to different variations in the environment dynamics, the resulting policy can better generalize to the dynamics of the real world.

Agent Training Process To optimize the injection process at the COSY accelerator, we employ a soft actor-critic (SAC) agent [7], which is an extension of the actor-critic method particularly suited for continuous action spaces. The SAC algorithm is a model-free, off-policy algorithm that optimizes a stochastic policy in an online setting. The key components of the algorithm are the actor neural network and the critic neural network. The actor network is responsible for generating the policy distribution, which defines the probability of selecting each action given a state. The critic network, on the other hand, estimates the expected return for each state-action pair. SAC is chosen for its sample efficiency and its ability to optimize the action entropy, which leads to better generalization and exploration in continuous action spaces.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA098

© Content from this work may be used under the terms of the CC BY 4.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

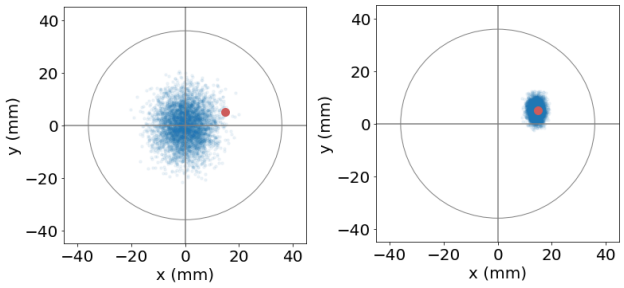

Figure 3: An optimization instance for the agent in simulation. The blue dots are the simulated particles and the red dot is the target. Both images resembles the state of the IBL at the injection point into COSY. The left figure illustrates the beam before the optimization and the right figure after the optimization by the RL agent.

The training process for the soft actor-critic agent involves the following steps:

1. Randomly sample the dynamic parameters ρ and the goal g_t at the beginning of each episode.
2. The agent observes the current state s_t and the goal g_t.
3. The agent selects an action a_t based on its current policy $\pi(a|s_t, g_t)$, which is optimized for continuous action spaces.
4. The agent receives a reward signal $r_t = r(s_t, a_t, g_t)$ from the environment, which incorporates the goal and the dynamic parameters.
5. The agent transitions to the next state s_{t+1}, determined by the environment dynamics $P(s_{t+1}|s_t, a_t, \rho)$.
6. Update the critic by optimizing the Q-function $q(s, a, g, \rho)$.
7. Update the actor by gradient ascent to increase the expected reward from the Q-function. The soft actor-critic agent also optimizes the action entropy to encourage better exploration and generalization.
8. Repeat steps 2-7 until convergence.

This approach allows the agent to effectively learn a policy that adapts to a continuous action space, achieves sample efficiency, and promotes exploration and generalization. Figure 3 displays a trained agent optimizing the beam at the injection point to match a dictated target in simulation. Incorporating domain randomization with this approach allows the agent to optimize the injection process for COSY, while accounting for the continuous nature of the action space and the complex dynamics of the environment.

PREPARATIONS FOR APPLICATION AT COSY

To demonstrate the transfer from simulation to real world a dedicated beam time at the COSY IBL has been requested and was granted by the COSY beam-time advisory committee CBAC [8]. The beam time benefits from the EPICS control system [9], which allows for automated control and read-out of accelerator parameters not only by operators but as well by algorithms. The EPICS control system for the IBL was introduced in late 2021.

To enable the topical Reinforcement Studies the hard- and software of the viewer at the end of the IBL have been modernised recently:

In order to record the beam cross section at the location of the charge exchange foil, the foil has to be moved out and instead a viewer screen has to be inserted at the same location. This process had to be performed manually at the location of the viewer. After the mentioned hardware upgrade this can now be triggered by the EPICS system and is then carried out automatically by a drive belt system. So both the image at the screen and influence of the manipulations of the agent on the injected beam at COSY can be measured successively without timely manual interactions in the COSY tunnel. Additionally the viewer image is now read by EPICS and directly analysed: As a result the centre of gravity as well as the width of the beam are now accessible to the user and hence the algorithms.

ACKNOWLEDGEMENT

Simulations were performed with computing resources granted by RWTH Aachen University under project rwth0905.

REFERENCES

[1] R. Maier *et al.*, "Cooler synchrotron COSY", *Nucl. Phys. A*, vol. 626, nos. 1-2, pp. 395–403, 1997. doi:10.1016/S0375-9474(97)00562-9

[2] U. Bechstedt *et al.*, "The cooler synchrotron COSY in Jülich" *Nucl. Instrum. Methods Phys. Res., Sect. B*, vol. 113, nos. 1-4, pp. 26–29, 1996. doi:10.1016/0168-583X(95)01352-0

[3] A. Awal, J. Hetzel, R. Gebel, V. Kamerdzhiev, and J. Pretz, "Optimisation of the injection beam line at the Cooler Synchrotron COSY using Bayesian Optimisation", *J. Instrum.*, vol. 18, no.4, p. P04010, 2023. doi:10.1088/1748-0221/18/04/P04010

[4] R. Sutton and A. Barto, *"Reinforcement Learning: An Introduction"*, MIT Press, Second Edition, Cambridge, MA, 2018.

[5] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world", in *Proc. 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 23–30. doi:10.1109/IROS.2017.8202133

[6] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators" *Proc. 32nd Int. Conf. on Machine Learning*, vol. 37, Lille, France, Jul. 2015, pp. 1312–1320, https://proceedings.mlr.press/v37/schaul15.htm

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor", *arXiv preprint*. doi:10.48550/arXiv.1801.01290.

[8] M. Weber *et al.*, "Minutes of the 14th Meeting of the COSY Beamtime Advisory Committee (CBAC)", GSI, Darmstadt, Germany, 23–24 Feb. 2023, p. 10, https://collaborations.fz-juelich.de/ikp/jedi/public_files/cbac_reports/minutes_CBAC14_v7.pdf

[9] Experimental Physics and Industrial Control System (EPICS), https://epics-controls.org/