ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST040

AUTOMATED INTENSITY OPTIMISATION USING REINFORCEMENT LEARNING AT LEIR

N. Madysa*, V. Kain†, CERN, Geneva, Switzerland
R. Alemany Fernandez, N. Biancacci, B. Goddard, F. M. Velotti, CERN, Geneva, Switzerland

Abstract

High intensities in the Low Energy Ion Ring (LEIR) at CERN are achieved by stacking several multi-turn injections from the pre-accelerator Linac3. Up to seven consecutive 200 µs long, 200 ms spaced pulses are injected from Linac3 into LEIR. Two inclined septa, one magnetic and one electrostatic, combined with a collapsing horizontal orbit bump allows a 6-D phase space painting via a linearly ramped mean momentum along the Linac3 pulse and injection at high dispersion. The already circulating beam is cooled and dragged longitudinally via electron cooling (e-cooling) into a stacking momentum to free space for the following injections. For optimal intensity accumulation, the electron energy and trajectory need to match the ion energy and orbit at the e-cooler section.

In this paper, a reinforcement learning (RL) agent is trained to adjust various e-cooler and Linac3 parameters to maximise the intensity at the end of the injection plateau. Variational Auto-Encoders (VAE) are used to compress longitudinal Schottky spectra into a compact latent space representation as state input for the RL agent. The RL agent is pre-trained on a surrogate model of the LEIR e-cooling dynamics, which in turn is learned from the data collected for the training of the VAE. The performance of the VAE, the surrogate model, and the RL agent is investigated in this paper. An overview of planned tests in the upcoming LEIR runs is given.

INTRODUCTION

LEIR [1, sec. 4] is an ion synchrotron equipped with an e-cooler. It is situated in the CERN accelerator chain between Linac3 and the Proton Synchrotron (PS).

In nominal operation mode, it receives ions from Linac3 at 4.2 MeV per nucleon in seven consecutive pulses per cycle. The pulses have a spacing of 200 ms and are 200 µs long. They are injected into LEIR via 6-D phase space painting with a collapsing horizontal bump and a momentum sweep. The latter is achieved with the debunching and the ramping radio-frequency (RF) cavity at the exit of Linac3.

While circulating, each pulse is cooled and longitudinally dragged by the e-cooler and stacked in a narrow phase space volume to make space for the next pulse. Once all pulses are injected, the beam is dragged back to nominal momentum, captured into bunches and accelerated to the nominal target beam rigidity. (For lead ions, this target corresponds to an energy of 72.2 MeV per nucleon.) It is then finally ejected into the transfer line towards the PS.

* nico.madysa@cern.ch
† verena.kain@cern.ch

Ten parameters have been identified as crucial for this process: the start and end phase of the ramping and the debunching RF cavity; the electron gun voltage at the start and end of e-cooling; as well as the orbit positions (x, y) and angles (x', y') at the e-cooler, controlled by orthogonal orbit bumps.

These parameters $\vec{p}$ require frequent adjustment to ensure that LEIR maintains its beam intensity in the ring after RF capture ($I_{\mathrm{R,cap}}$) above the nominal target $I^{\mathrm{nom}}_{\mathrm{R,cap}} = 10 \times 10^{10}$ charges. Since LEIR entered operation in 2005, these parameters have been optimised manually by the operations team. This is time consuming and often based on trial and error. Instead, we propose an automatic system based on machine learning that maximises beam intensity in a fast and deterministic manner.

Data acquisition at LEIR takes several seconds per machine interaction. This precludes approaches that require many evaluations of the loss function, particularly gradient-based optimisation and model-free RL. Instead, we present an approach based on a *surrogate* model. Using a limited amount of data, we use supervised learning to train a model of LEIR's injection process and its dynamics at flat bottom, then optimise this surrogate numerically and via RL.

DATA TAKING

Two runs of data were taken at the end of 2021 under nearly identical machine conditions. Run 1 occurred on November 8 and took 4293 samples over 15.5 h, Run 2 on November 13 and took 4414 samples over 18 h.

Each sample recorded the following features:

- the parameters before ($\vec{p}_{n-1}$) and after the change ($\vec{p}_n$);
- the change itself, $\Delta\vec{p}_n := \vec{p}_n - \vec{p}_{n-1}$;
- the longitudinal Schottky spectrum after the change;
- the beam intensity after Linac3 and in the LEIR ring.

For both runs, the machine was first set up to the same parameters, which had been found by manual optimisation. This was recorded as the first sample. Then, for the rest of the run, parameter settings were sampled from a uniform distribution. They were applied to the machine and a new sample was recorded.

Samples were discarded if their settings violated safety limits. If the Linac3 intensity dropped below a certain threshold due to temporary faults, recording was temporarily suspended until it returned to expected levels.

During Run 1, parameters were sampled from a comparatively narrow distribution around the optimum; for Run 2, the distribution was widened and almost all samples had a very low intensity. This ensured that the data set contained a comparable amount of high- and low-intensity samples.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST040

SURROGATE MODEL

Interpreting Schottky Spectra

To train RL algorithms effectively, adequate state information is necessary. To this effect, a β-variational autoencoder [2] compresses each collected Schottky spectrum into a small *latent* vector z with only the most important information. The goal of a β-VAE is to reproduce its input as closely as possible. The loss term to minimise is

$$L_{\text{tot}}(x) = L_{\text{MSE}}(D(E(x)), x) + \beta D_{\text{KL}}(E(x)), \qquad (1)$$

where x is the Schottky spectrum, L_{MSE} is the mean squared error, D and E are the decoding and encoding half of the VAE, D_{KL} is the Kullback–Leibler divergence to a standard normal distribution $\mathcal{N}$, and β is its weight factor. By forcing the latent representation $z = E(x)$ to follow the distribution $\mathcal{N}$, it is possible to generate artificial Schottky spectra by evaluating $D(z \sim \mathcal{N})$.

A good VAE can be found in two steps: first, β is set to zero and the number of latent variables N_z is varied to find the smallest value that gives a nearly minimal L_{MSE}. Then, N_z is held constant and β is varied to minimise D_{KL} while keeping L_{MSE} low and the latent variables uncorrelated. Overfitting is prevented via early stopping. The data is split 90–5–5, meaning that 5 % are set aside for online validation and another 5 % for post-training evaluation.

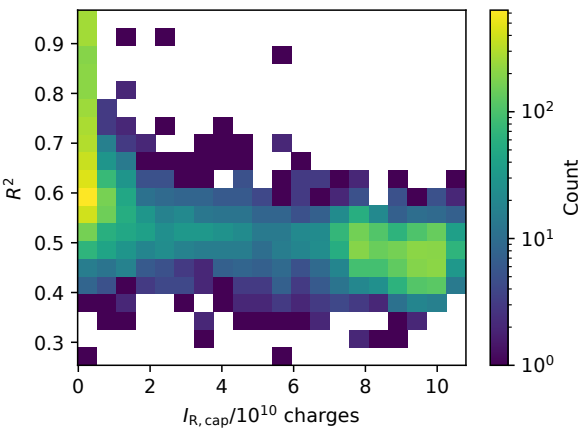

Figure 1: The collected data over beam intensity $I_{\text{R,cap}}$ and coefficient of determination R^2.

The final VAE uses $N_z = 12$ and $\beta = 0.005$. The coefficient of determination is $R^2 = 0.59 \pm 0.14$, and is slightly higher for low- than for high-intensity samples, as Fig. 1 shows. An example of its output is shown in Fig. 2. The latent variables have a mean $\mu_z = -0.02 \pm 0.04$, a standard deviation $\sigma_z = 0.92 \pm 0.10$ and are almost uncorrelated.

Intensity Model

The next step is to associate each latent vector z with our figure of merit, the beam intensity $I_{\text{R,cap}}$. For this task, a multi-layer perceptron (MLP) has been trained and evaluated on another 90–5–5 split of the data.

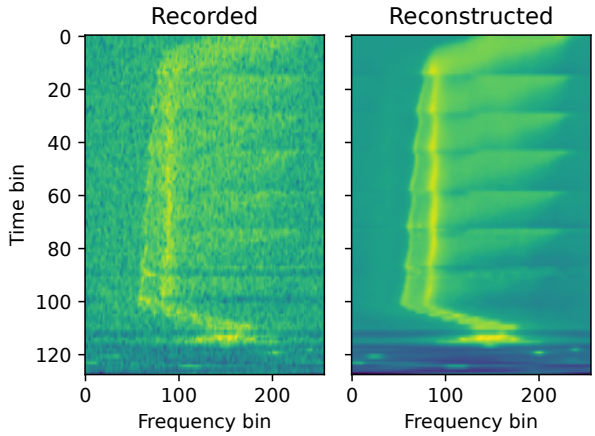

Figure 2: A Schottky spectrum from the training data set (left) and its reconstruction by the β-VAE (right).

The resulting model has a prediction error (mean and standard deviation) of $\left\langle I^{\text{pred}}_{\text{R,cap}} - I^{\text{true}}_{\text{R,cap}} \right\rangle = (-0.1 \pm 0.9) \times 10^{10}$ charges, which is an acceptable amount. The median absolute error is 0.2×10^{10} charges, which indicates that the mean is dominated by few samples with large errors.

Dynamics Model

The last piece is a model of the system dynamics. For this purpose, an MLP has been trained on the mapping $(\vec{z}_{n-1}, \Delta\vec{p}_n) \mapsto \vec{z}_n$. Parameter *changes* are used instead of absolute values in order to maintain the process' Markov property despite subtle effects that cause the machine's optimal working point $\vec{p}^*$ to change over time (*drifting*).

The recorded dataset under-samples the region of small parameter changes, $\|\Delta\vec{p}\| \approx 0$, a common problem in high-dimensional phase spaces. Because small $\Delta\vec{p}$ should lead to small changes in $\vec{z}$, the original dataset is concatenated with a "zero-action" dataset (mapping $(\vec{z}_n, 0) \mapsto \vec{z}_n$) of the same size. Again, a 90–5–5 split has been used for training. The evaluation is performed separately on the unmodified and the zero-action portion.

The trained model shows near-perfect performance on the zero-action dataset: The correlation coefficient c_i between predicted and target values is close to one for all z_i. For the real data, the c_i average at 0.54 and are spread nearly uniformly in the range $[0.25, 0.90]$. A principal-component analysis shows that generally, components with higher importance also are predicted with higher accuracy.

PARAMETER OPTIMISATION

Using these models, an optimisation problem has been constructed that is a surrogate of the real machine. It uses the CERN Common Optimisation Interfaces (COI) [3] for compatibility reasons.

For numerical optimisation, BOBYQA [4, 5] has been chosen as a black-box optimiser that operates on bounded spaces with few function evaluations. For each episode, an

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST040

initial $z_0 \sim \mathcal{N}$ is sampled. The goal is to find a Δp^* that minimises $-I_{\mathrm{R,cap}}$ as predicted by the surrogate model. Optimisation is done once with and once without global minimisation. Both variants are evaluated over 1000 episodes.

For RL, three algorithms have been used: SAC, PPO and TD3, all from the Stable Baselines 3 package [6]. They are trained for 1 000 000 steps and evaluated over 10 000 episodes. TD3 and SAC use normal-distributed action noise with $\mu = 0$, $\sigma = 0.1$.

The episodes work similarly to BOBYQA, except that the RL agents adjust the parameter change $\Delta\vec{p}$ with a series of increments $\Delta(\Delta\vec{p}_n)$ to find a point at which $I_{\mathrm{R,cap}} > I^{\mathrm{nom}}_{\mathrm{R,cap}}$. This avoids using the dynamics model iteratively, which would accumulate prediction errors and become unstable.

The reward $r_n = I^{(n)}_{\mathrm{R,cap}} - I^{\mathrm{nom}}_{\mathrm{R,cap}}$ is always negative, except for steps that end an episode successfully. Episodes also end after 20 steps or if $\|z\|_2^2$ grows beyond the domain on which the intensity model has been trained.

RESULTS

Table 1: Evaluation Results. P is the Fraction of Episodes in which an Algorithm has Reached the Target $I^{\mathrm{nom}}_{\mathrm{R,cap}} = 10 \times 10^{10}$ Charges. L is the Mean Number of Steps per Episode, ΔL its Standard Deviation.

Algorithm	P/%	$L \pm \Delta L$
BOBYQA (Global)	46.3	2300 ± 500
BOBYQA	40.8	76 ± 18
SAC	36.7	15 ± 7
PPO	35.7	16 ± 6
TD3	33.3	15 ± 7

The results are summarised in Table 1. The highest success rate is achieved by the globally optimising BOBYQA. Classic BOBYQA shows slightly worse performance, but cuts the number of function calls by a factor of 30. The RL algorithms all show moderately worse performance while cutting episode length by an additional factor of 4.5.

As Fig. 3 shows, the success rate depends greatly on the beam intensity before optimization. In cases where the machine is only slightly mistuned ($I^{\mathrm{init}}_{\mathrm{R,cap}} > 8 \times 10^{10}$ charges), all algorithms perform about equally well with a success rate of $P \approx 95\,\%$. It also shows that most of the unsuccessful episodes are those for which almost no beam is injected.

Figure 4 shows that BOBYQA restores some amount of intensity in almost all cases. By contrast, the RL agents end about 20 % of episodes with almost no intensity. They also show a sharp drop-off at $I^{\mathrm{nom}}_{\mathrm{R,cap}}$ because this is where their episodes are terminated in any case.

CONCLUSION

A reasonably accurate surrogate model of the LEIR machine has been trained using a β-VAE and two MLPs.

We find that BOBYQA recovers some amount of beam in almost all cases, even if almost no beam is captured after injection. Global optimisation performs slightly better than the default, but becomes prohibitively long on the real machine, where each data acquisition takes several seconds.

Conversely, if the machine is already close to the nominal target intensity, all algorithms perform equally well. In this case, RL algorithms reach the target in fewer than 20 steps, whereas BOBYQA needs 21 steps for its bootstrapping phase alone.

During LEIR commissioning in the summer of 2022, we plan to compare the surrogate model to the machine, verify the BOBYQA results (in particular the success rate) and evaluate the RL agents that have been trained off-line. Another promising avenue to explore are model-based RL agents that are trained on-line on the machine. The VAE presented here will remain useful there, as it provides a means to extract meaningful state information from the observed Schottky spectra.

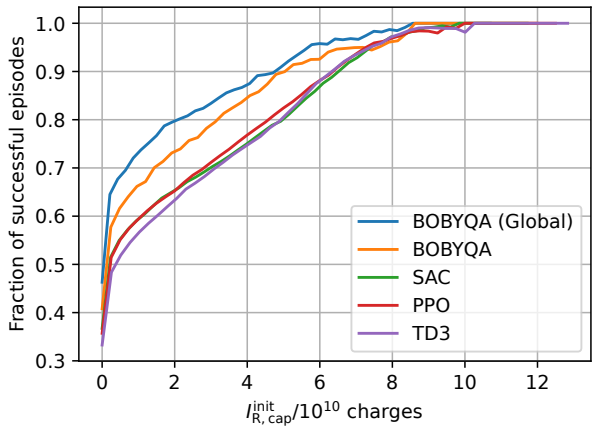

Figure 3: Success rate of the various algorithms over initial beam intensity $I^{\mathrm{init}}_{\mathrm{R,cap}}$. For each point, this selects evaluation episodes with at least this $I_{\mathrm{R,cap}}$ before optimization and shows the fraction of them in which $I^{\mathrm{nom}}_{\mathrm{R,cap}}$ has been reached.

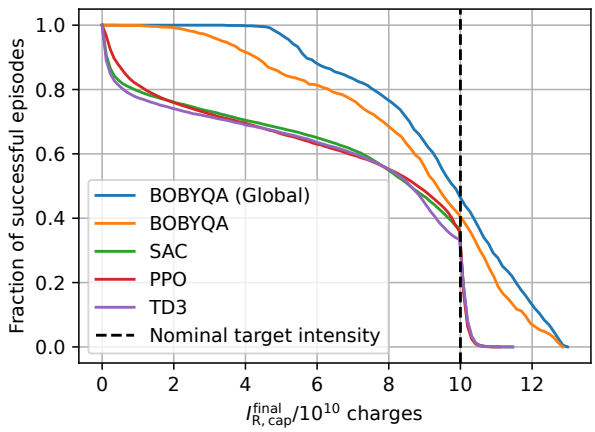

Figure 4: ROC curve of the various algorithms. It shows the fraction of the evaluation episodes in which at least the given $I_{\mathrm{R,cap}}$ has been reached.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-TUPOST040

REFERENCES

[1] J. Coupard, H. Damerau, A. Funken *et al.*, "LHC injectors upgrade, vol. 2", CERN, Geneva, Switzerland, Tech Rep. CERN-ACC-2016-0041, Apr. 2016. doi:10.17181/CERN.L6VM.UOMS

[2] C. P. Burgess, I. Higgins, A. Pal, L. Matthey *et al.*, "Understanding disentangling in β-VAE", Apr. 2018. doi:10.48550/ARXIV.1804.03599

[3] Common optimization interfaces documentation, https://cernml-coi.docs.cern.ch

[4] C. Cartis, J. Fiala, B. Marteau and L. Roberts, "Improving the flexibility and robustness of model-based derivative-free optimization solvers", *ACM Trans. Math. Softw.*, vol. 45, no. 3, pp. 1-41, Sep. 2019. doi:10.1145/3338517

[5] C. Cartis, L. Roberts and O. Sheridan-Methven, "Escaping local minima with local derivative-free methods: a numerical investigation", *Optimization*, pp. 1-31, 2021. doi:10.1080/02331934.2021.1883015

[6] A. Raffin, A. Hill *et al.*, "Stable-baselines3: reliable reinforcement learning implementations", *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1-8, Nov. 2021.