Batch Spacing Optimization at SPS injection by Reinforcement Learning

M. Remta, F. Velotti, CERN, Geneva, Switzerland
S. Rezagholi, UAS Technikum, Vienna, Austria

DOI:10.1103/g9wr-197z

Motivation

- Batches of particles are sequentially injected into CERN's Super Proton Synchrotron (SPS) with a spacing of 200 ns.
- The injection kickers, which are controlled by eight delays, have to be well synchronized with the beams.
- Trade-off between perturbation of the circulating beam and insufficient deflection of the injected beam due to kicker rise time, causing injection oscillations and degrading the beam quality.
- Present solution: numerical optimization (BOBYQA [1]). Requires to be rerun after drifts of the system and has to learn the problem anew each time.
- Reinforcement Learning was investigated as alternative, aiming for faster optimization and an active controller.

Methods

Environment

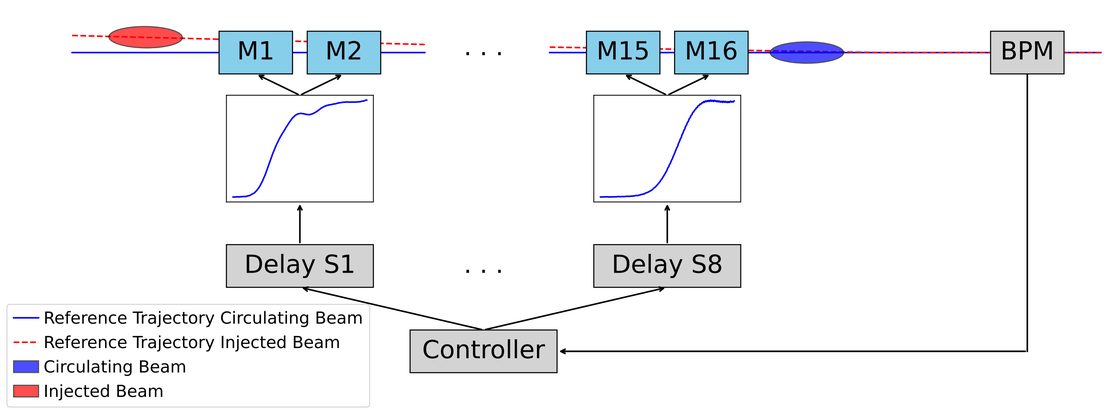

FIG. 1. Schematic of the environment. The 16 magnet modules (M1 - M16) are powered in pairs of two. Each circuit is controlled by the delay of an electrical switch (S1 - S8). The objective is to minimise deviations from the reference trajectory at a beam position monitor (BPM) downstream.

Model

- Reinforcement-Learning agent based on the PPO algorithm [2].
- An LSTM-network was integrated into the policy of PPO, enabling it to leverage information of the optimization history (see Fig. 2).
- Trained offline for 10^6 steps.

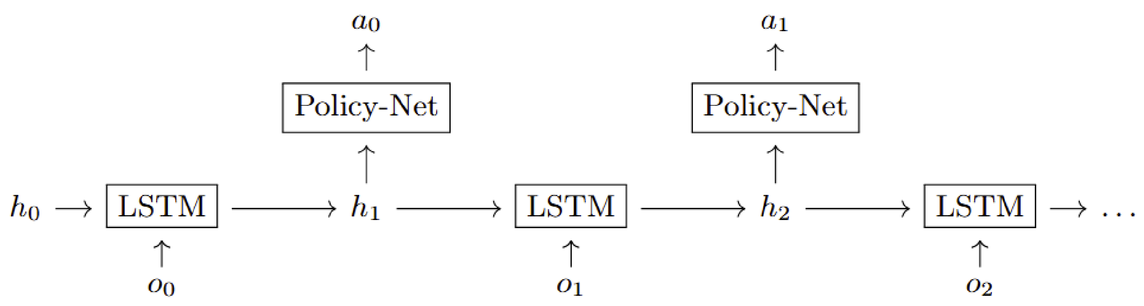

FIG. 2. Schematic of recurrent architecture for PPO. The initial internal state of the LSTM (h_0) is a zero vector.

Simulations

- After hyperparameter-tuning, the best RL-agent was benchmarked against the numerical optimizer BOBYQA over 10,000 episodes (see Fig. 3).
- The agent required 7 steps in the median for the optimization, BOBYQA 246.
- Final loss: on average, the agent converged to a 2.73% higher loss than the numerical optimizer.

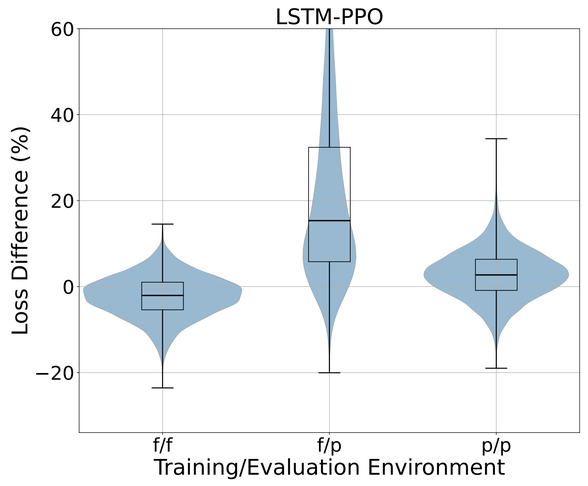

FIG. 3. Boxplots of relative differences in loss between RL-agent and BOBYQA. Training/Evaluation environment. f = fully-observable, p = partially-observable.

Accelerator

Dedicated study (2024)

- Agent was entirely trained offline and directly deployed to the accelerator.
- Two evaluation episodes, starting from optimized initial settings (see Fig. 4).
- Random actions taken throughout the episodes (dashed lines in Fig. 4), imitating perturbations.
- Agent aims to reach and maintain a specific state, which can be seen from the trajectory of aggregated actions.

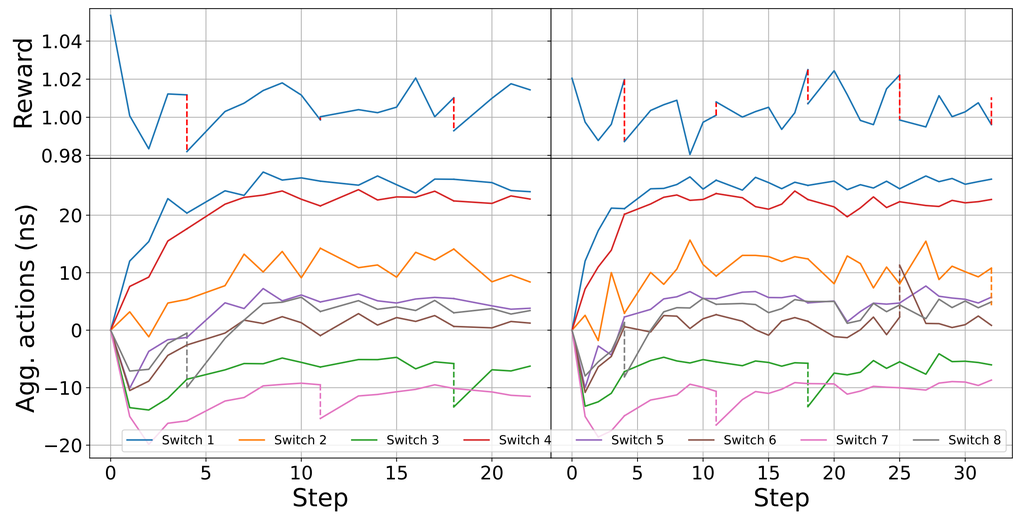

FIG. 4. Performance of agent during the two trials on the accelerator. Up: Rewards during the trial. Down: Cumulative changes in delays. The dashed vertical lines indicate a random perturbation.

SPS commissioning (2026)

- Agent significantly improved upon the initial state and found good relative delays between the individual waveforms (see Fig. 5).
- After convergence, the oscillations of the circulating beam were still larger in amplitude than the injected beam.
- By manually applying a common shift to all the waveforms, the oscillations amplitudes were balanced.

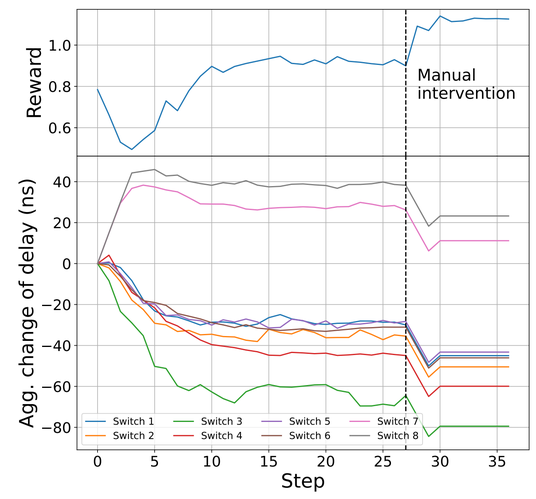

FIG. 5. Performance of agent during commissioning.

Summary and Outlook

- A robust simulation environment, incorporating variations such as different waveforms, random changes in rise time, and variability in initial delays, is essential to obtain effective agents.
- Partially observable Markov decision problem due to noise and errors.
- PPO with memory (LSTM) proved effective for this problem.
- In simulations, the agent optimized much faster than BOBYQA but reached slightly worse final losses.
- On the accelerator, the agent showed clear intention to reach and maintain a state close to the optimum, highlighting its capabilities as an active controller.
- Two years after initial training, the agent still performed well.
- Common translations in time of the waveform information remain an issue.

KEY REFERENCES

[1] Powell, M. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report, University of Cambridge, Dept. of Applied Mathematics and Theoretical Physics.

[2] Schulman, J., Wolski, F., Dhariwal, P., Radford A., and Klimov O. (2017). Proximal policy optimization algorithms. ArXiv:1707.06347.

[3] Remta, M., Velotti, F., and Rezagholi, S. (2025). Batch spacing optimization by reinforcement learning. Phys. Rev. Accel. Beams, 28(9), 094603

ACKNOWLEDGEMENT AND PARTNERS

CERN
FH University of Applied Sciences TECHNIKUM WIEN

MORE INFORMATION

Matthias Remta
CERN – SY/ABT/BTP
matthias.remta@cern.ch
https://gitlab.cern.ch/mremta/mkp-delays-rl