ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL038

ULTRA FAST REINFORCEMENT LEARNING DEMONSTRATED AT CERN AWAKE

S. Hirlaender*, L. Lamminger, Paris Lodron University Salzburg, Austria
Z. Della Porta, V. Kain[1], CERN, Geneva, Switzerland

Abstract

Reinforcement learning (RL) is a promising direction in machine learning for the control and optimisation of particle accelerators since it learns directly from experience without needing a model a-priori. However, RL generally suffers from low sample efficiency, and thus training from scratch on the machine is often not an option. RL agents are usually trained or pre-tuned on simulators and then transferred to the real environment. In this work, we propose a model-based RL approach based on Gaussian processes (GPs) to overcome the sample efficiency limitation. Our RL agent was able to learn to control the trajectory at the CERN AWAKE (Advanced Wakefield Experiment) facility, a problem of 10 degrees of freedom, within a few interactions only. To date, numerical optimises are used to restore or increase and stabilise the performance of accelerators. A major drawback is that they must explore the optimisation space each time they are applied. Our RL approach learns as quickly as numerical optimisers for one optimisation run, but can be used afterwards as single-shot or few-shot controllers. Furthermore, it can also handle safety and time-varying systems and can be used for the online stabilisation of accelerator operation. This approach opens a new avenue for the application of RL in accelerator control and brings it into the realm of everyday applications.

GENERAL PROBLEM DESCRIPTION

RL holds tremendous promise in controlling accelerators. Still, everyday applications are rare. Several reasons can be identified:

- Sample efficiency
- State space observability
- Safety constraints
- Non-stationarity

Several approaches have been applied to mitigate the sampling efficiency problems [1,2]. When the reward objective has a horizon of one step, the RL problem is reduced to optimising parameters to maximise the objective greedily. Therefore, classical numerical optimisers can generally be used, providing robust and fast results. A major drawback is that they must learn the problem from scratch each time they are used. In this paper, we present a model-based RL method based on [3] that learns the control problem from scratch as fast as optimisers. The approach successfully solves three challenges of the four challenges, as demonstrated in experiments. The test benchmark was the electron line of the AWAKE experiment at CERN, as described in the following section.

* simon.hirlaender@plus.ac.at

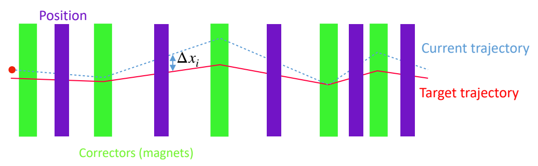

Figure 1: Illustration of a beam steering problem as in the AWAKE electron line.

PROBLEM DEFINITION - AWAKE ELECTRON LINE TRAJECTORY STEERING

The electron line of AWAKE (see Fig. 1) served in the past as an excellent environment to test optimisation and control algorithms, as also an accurate simulation of the electron beam in the line is available. The trajectory steering problem is episodic with the goal to minimise the distance of an initial beam trajectory to a target trajectory as quickly as possible. Ten dipole magnets can be changed to steer the beam defining the actions **a**, and ten beam position monitors measure the trajectory defining the state **s**. The reward r is the negative RMS value of the distance to the target trajectory. If a threshold RMS (-1 mm in our case) is surpassed, the episode ends successfully. If the beam hits the wall (any state ≤ -1 cm or ≥ 1 cm), the episode is terminated unsuccessfully. All episodes are initialised such that the RMS of the distance to the target trajectory is between 0.7 cm and 0.8 cm, where all states and actions are normalised to $[-1, 1]$. This ensures that the task is not too easy and relatively close to the boundaries to probe the safety settings.

METHOD

To address the problem, a model-based RL technique is employed (see Fig. 2). The transition dynamics and reward model are approximated using an uncertainty-aware data-driven model based on GPs, which are known for high sample efficiency. No initial knowledge (except the init kernel and scaling) is given, and the algorithm learns - following the RL paradigm - to solve the problem through trial and error from scratch. After a few initialisation steps, the model is trained, and an action sequence for T time steps in the future is optimised on the model. The first step is executed, and afterwards, the new data is used to improve the model, and again the optimal action sequence for T time steps in

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-THPL038

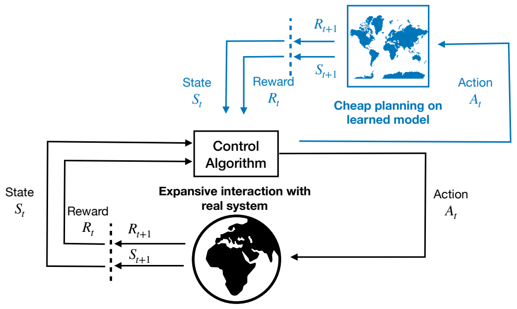

Figure 2: The concept of model-based RL.

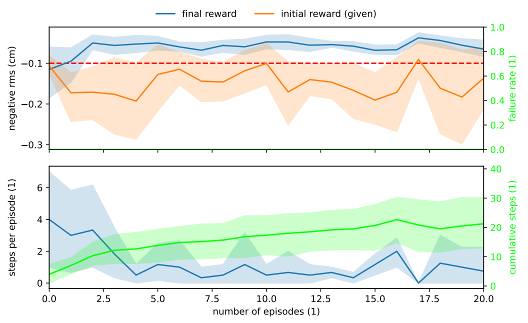

Figure 3: Results on the real machine.

the future is searched. This procedure continues until a sufficiently good model has been achieved. The approach is close to methods used for model predictive control [4]. The model has the following structure:

$$\mathbf{p}_{t+1}(\mathbf{s}_t, \mathbf{a}_t) = f(\mathbf{s}_t, \mathbf{a}_t) + t, \tag{1}$$

where $_t \sim \mathcal{N}(\mathbf{0},)$ and $\mathbf{p}_t := (\mathbf{s}_{t+1}, r_t)$. Our aim is to find a policy $\pi^*_t(\mathbf{s}_t) \mapsto \mathbf{a}_t$, which maximises the expected mean reward, where the reward objective is:

$$\pi^*_t = \max \pi_t \lim_{T\to\infty} \mathbb{E}[\sum_{t=0}^{T} \frac{1}{T} r_t] \tag{2}$$

$$\text{subject to } \mathbf{s}_{t+1}, r_t \sim \mathbf{p}_{t+1}(\mathbf{s}_t, \mathbf{a}_t) \tag{3}$$

$$\mathbf{a}_t = \pi_t(\mathbf{s}_t) \tag{4}$$

$$\mathbf{s_0} \sim \rho_0 \text{ (given initial distribution).} \tag{5}$$

Via moment matching [5], it is possible to propagate the expected states and the corresponding uncertainties in a closed-form, which allows for efficient optimisation of Eq. 2 using the gradient and hessian information. Consequently, closed forms are also available for the violation probability of safety constraints on the states. To be able to use the episodic training setting, as explained in the previous section, we do not consider data when resetting an episode (which can be interpreted as a perturbation of the system).

Transition and Reward Model

The transition model is captured via GPs. The state transitions are important to model the uncertainty propagation in

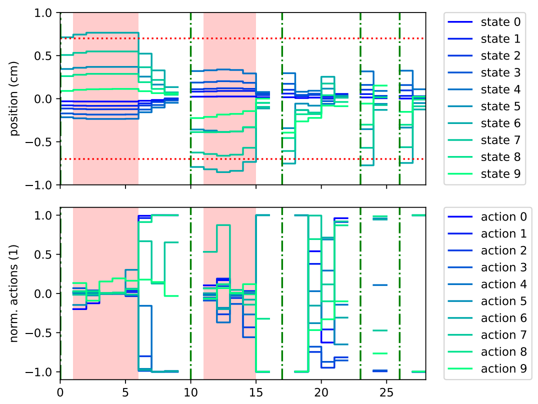

Figure 4: The policy evolution during the training highlighting (red) the safety policy. Vertical green lines show the reset of the episodes. All episodes were successfully finished.

order to estimate the risk of violating the safety constraints. The reward model is trained on an external reward signal, which is often demanded in real-world scenarios. If the reward is formulated as a function of the states and actions, the learning of a reward function is unnecessary. Hence, the sample complexity can be improved (used in the non-stationarity experiments).

The Reward Objective

Instead of solving Eq. 2, we solve the finite horizon problem in an iterative way, as explained previously. The reward model yields the expected RMS and the uncertainty, which can be used to find a trade-off between exploitation and exploration. Several experiments were conducted with varying exploration, but in the presented results, only the upper confidence bound of the estimated RMS was used. Since the AWAKE benchmark can be solved with a short horizon, due to the convexity of the reward w.r.t. the states and actions, we use a horizon of length one.

Safety Constraints

The uncertainty of a potential violation is propagated analytically and treated in a constrained optimisation up to the second order using the 'trust-constr' optimisation of Scipy [6]. Additionally, the proposed steps are scaled inversely proportionally to the estimated risk of hitting the boundaries as delivered by the model. In our tests, the initial settings are relatively close to the boundary to challenge the risk awareness of the algorithm. In all other experiments, the Scipy 'L-BFGS' optimiser up to the second order is used.

Time Varying Systems

Time-varying systems are partially observable Markov decision processes [7], where one or several unobserved parameters change the response of the system slowly in time, like drifts of the magnets due to slow temperature changes. To be able to learn to control the system in this considerably harder case, we provide the time step as extra input to the model to consider more recent data to be more important.

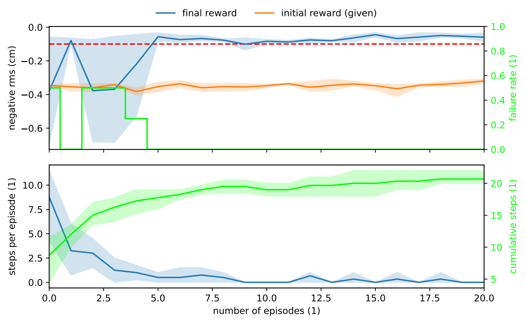

Figure 5: Simulation without safety settings.

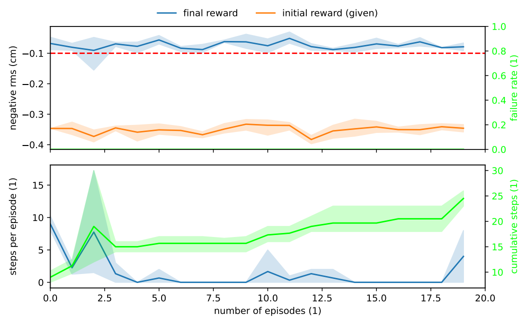

Figure 6: Simulation with safety settings.

EXPERIMENTS AND RESULTS

Several experimental campaigns were carried out, including a test on a real machine to verify the results directly. In the following, we show the results of these studies. In all experiments, five independent learning procedures are averaged, and the standard deviation is shown as shaded area. Five initial state-action-next-state reward transitions are deterministically chosen in each case to homogenise the experimental settings.

Sample Complexity on AWAKE

In Fig. 3, the results of the tests on the real machine are shown. These results were obtained with the settings found in the simulations. The algorithm did not violate the constraints within the experiment campaign without the safety settings and learned to control the system within several steps only.

Safety Constrains

Cases were simulated where safety could be critical and compared to cases where risk was not considered. If the probability that one of the predicted states exceeds a certain threshold is higher than a certain threshold calculated by the constrained optimization, the proposed action is reduced inversely proportionally to this probability. This is done to ensure safe actions, as shown in Fig. 4 for a test where the reduction of unsafe states is indicated as a red-shaded area. The horizontal red lines display the safety limits of the states. With this setting, the safety constraints were not violated (see Fig. 6), while without this setting, they are violated (see Fig. 5).

Time Varying Systems

To model a non-stationary situation, we built a simulation where the quadrupole strengths of the AWAKE environment change sinusoidally with a frequency of 0.01 and an amplitude of 0.5 per time step. The stationary approaches fail in such a setting as shown in Fig. 7. It was, however, possible to successfully handle such cases by switching to the time-dependent setting and even considering safety, see Fig. 8.

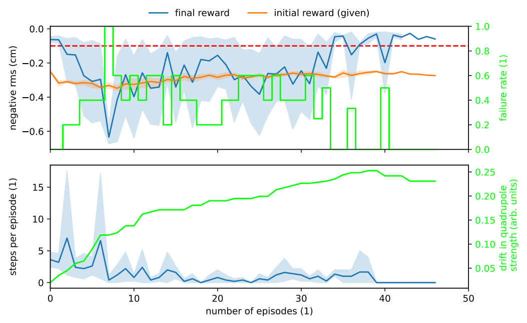

Figure 7: Non-stationary case fails using the stationary approach.

CONCLUSION

We have successfully overcome the challenges of sampling efficiency, safety, and non-stationarity of RL by using a model-based approach using Gaussian Processes. Studies were performed on the AWAKE electron line in simulations, as well as tests on the real machine to confirm the results. More challenging problems require a longer planning horizon, leading to various complications, such as increased computation time of the optimisation step, which will be examined in further studies. The results are promising and pave the way for further applications of RL in the domain of accelerator control.

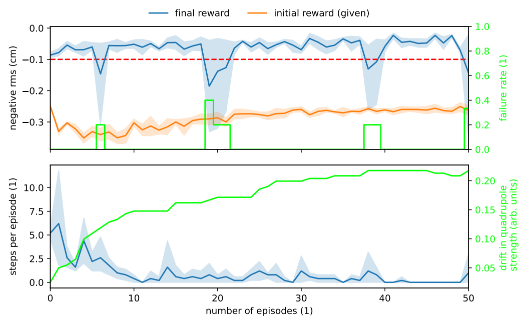

Figure 8: Non-stationary case with safety constraints and time dependency of the GPs.

REFERENCES

[1] V. Kain *et al.*, Sample-efficient reinforcement learning for CERN accelerator control", Phys. Rev. Accel. Beams, 23.124801 (2020).

[2] V. Kain *et al.*, "Test of Machine Learning at the CERN LINAC4", in *Proc. HB'21*, Batavia, IL, USA, Oct. 2021, pp. 181–185. `doi:10.18429/JACoW-HB2021-TUEC4`

[3] S. Kamthe *et al.*, Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control", PMLR 84:1701-1710, *Twenty-First International Conference on Artificial Intelligence and Statistics*, PMLR 2018.

[4] F. Borrelli *et al.*,Predictive Control for Linear and Hybrid Systems."*Cambridge University Press*, 2017.

[5] A. Girard *et al.*, Gaussian Process Priors with Uncertain Inputs-Application to Multiple-Step Ahead Time Series Forecasting", *Advances in Neural Information Processing Systems*, 2003.

[6] P. Virtanen *et al.*,SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, vol. 51, 2020.

[7] M. Puterman, "Markov Decision Processes: Discrete Stochastic Dynamic Programming", *Wiley Series in Probability and Statistics*, 1994.