ACCELERATING MACHINE LEARNING FOR MACHINE PHYSICS (AN AMALEA-PROJECT AT KIT)

T. Boltz and W. Wang*, E. Bründermann, A. Kopmann, W. Mexner, A.-S. Müller
Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

The Innovation Pool project Amalea of the Helmholtz association of Germany will explore and provide novel cutting-edge machine learning techniques to address some of the most urgent challenges in the era of large data harvests in physics. Progress in virtually all areas of accelerator-based physics research relies on recording and analyzing enormous amounts of data. This data is produced by progressively sophisticated fast detectors alongside increasingly precise accelerator diagnostic systems. As KIT contribution to Amalea, it is planned to investigate the design of a fast and adaptive feedback system that reacts to small changes in the charge distribution of the electron bunch and establishes extensive control over the longitudinal beam dynamics. As a promising and well-motivated approach, reinforcement learning methods are considered. In a second step the algorithm will be implemented as a pilot experiment to a novel PCIe FPGA readout electronics card based on ZYNQ UltraScale+ MultiProcessor System on-Chip (MPSoC).

INTRODUCTION

With the increasing demand for compact, energy- and cost-efficient accelerator systems, in addition to tailored photon emission matched to the often extreme requirements of experiments in physics and photon science, the control systems have to cope with increasing complexity, high sensor data output rates, large data volumes as well as the desire for fast feedbacks and extensive beam control. Artificial intelligence with its subfield of machine learning including unsupervised, supervised and reinforcement learning, as well as deep learning, promises to assist in reducing the effort and complexity for operating a control system up to the point, where it may eventually control an accelerator autonomously. At the Karlsruhe Institute of Technology (KIT), since a few years, we are exploring machine learning methods for data classification, data reduction, and accelerator control informed by fast and precise sensor networks [1–4]. Since 2019, the Helmholtz Association in Germany is funding an Innovation Pool project called Amalea (Accelerating Machine Learning for Physics), which is exploring machine learning for accelerator-based physics, fast data reduction, and fast feature extraction from data, to name a few application areas. Amalea is driven by four Helmholtz centers, led by Deutsches Elektronen-Synchrotron (DESY), Helmholtz-Zentrum Berlin (HZB), Helmholtz-Zentrum Dresden-Rossendorf (HZDR) and KIT. The aim of Amalea is to investigate how novel machine learning methods, applied to the fields of particle physics, photon science and accelerator physics provide meaningful and effective use cases. At KIT and as one of the use cases contributing to the Amalea project, we explore how we can accelerate machine learning algorithms in real-time for machine physics applications and control. In this contribution, we discuss our efforts towards the design of a longitudinal feedback that acts on the RF system of the KIT storage ring KARA (Karlsruhe Research Accelerator) and aims for control of the micro-bunching instability. Driven by the interaction of short electron bunches with their own emitted coherent synchrotron radiation (CSR), this instability leads to the formation of dynamically changing microstructures within the longitudinal charge distribution of the bunch. Given its dynamic nature, a fast and adaptive feedback system is required to establish extensive control over the longitudinal beam dynamics. Reinforcement learning is a general-purpose approach to solving such problems, which has seen great success over the past decades. In [4], we illustrate how reinforcement learning can be applied to this task specifically, yielding the design of a longitudinal feedback loop. In the following, we review this idea and, in extension to [4], discuss some of the challenges in implementing this approach on a fast hardware system to meet the strict requirements regarding execution time. Therefore, KIT is developing a reinforcement learning hardware platform for the eventual implementation of the feedback design discussed below. The platform consists of two boards, the KAPTURE-2 front-end electronics that samples the pulse from the accelerator, and a high-end FPGA data acquisition board that provides high-data volume throughput that can process the data continuously. Based on which, a fast neural network inference can be deployed on FPGA for the fast inference requirement, and a lightweight training process is developed on ARM (or both on ARM side). To provide a proof of concept, the textbook *CartPole* environment is built on a ZYNQ MPSoC platform to test the performance of the reinforcement learning algorithm on hardware.

MICRO-BUNCHING INSTABILITY

Above a certain threshold current, which depends on the machine settings of the storage ring [5], the CSR self-interaction of short electron bunches leads to a dynamically changing longitudinal charge distribution and thus to fluctuating CSR emission (illustrated in Fig. 1). These fluctuations have been measured at a wide range of synchrotron light sources [6–18]. Additionally, the underlying longitudinal dynamics can be simulated by numerically solving the Vlasov-Fokker-Planck (VFP) equation [19], where the CSR

* These authors contributed equally to the presented work.

wake potential

$$V_{\mathrm{CSR}}(q) = \int_{-\infty}^{\infty} \widetilde{\rho}(\omega) Z_{\mathrm{CSR}}(\omega) e^{i\omega q} \mathrm{d}\omega \,, \qquad (1)$$

can be added as a perturbation to the Hamiltonian. Here, $q = (z - z_s)/\sigma_{z,0}$ denotes the generalized longitudinal position, $\widetilde{\rho}(\omega)$ the Fourier transformed longitudinal bunch profile and $Z_{\mathrm{CSR}}(\omega)$ the CSR-induced impedance of the storage ring. At the KIT storage ring KARA, such simulations using the VFP solver Inovesa [20] have shown great qualitative agreement with measurements of the emitted CSR power [21].

The additional potential in Eq. (1) can be interpreted as a perturbation to the accelerating RF potential, and thus results in a perturbation of the synchrotron motion within the bunch. This causes the formation of micro-structures and their dynamic evolution at time scales comparable to the synchrotron period. As the longitudinal charge distribution varies, so does the emitted CSR power, which is why this phenomenon is commonly referred to as micro-bunching or microwave instability. This also means that any efforts towards stabilizing the CSR emission imply obtaining some form of control over the micro-bunching dynamics within the bunch. However, depending on the application, the formation of such micro-structures can also be desirable as it leads to the emission of CSR at higher frequencies, reaching up to the THz range. Extensive control over the longitudinal beam dynamics would thus provide the opportunity of optimizing the emitted CSR for each application individually.

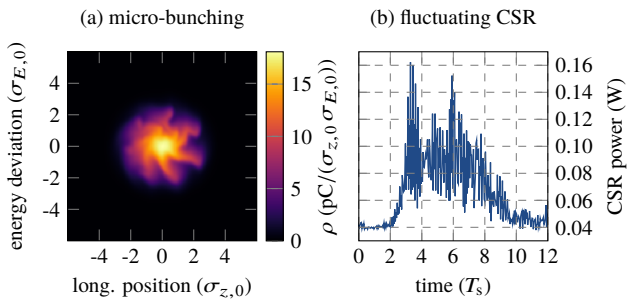

Figure 1: (a) The CSR self-interaction causes the formation of micro-structures in the longitudinal charge distribution. (b) Their continuous variation leads to fluctuations in the emitted CSR power. The illustrated dynamics are simulated with the VFP solver Inovesa developed at KIT.

REINFORCEMENT LEARNING

Following the notation in [22], the field of reinforcement learning (RL) is briefly introduced below. For a more detailed description of the subject, we refer to [22].

Reinforcement learning is the computational approach to goal-directed learning from interaction. In contrast to other sub-fields of machine learning, its learning paradigm does not require a pre-existing data set, but merely an *environment* to interact with. The learner and decision maker, usually called the *agent*, continuously interacts with the environment learning from past experience and thereby improving its behavior. At every time step, the agent perceives the current *state* S_t of the environment and carries out an *action* A_t. Based on the chosen action, the environment transitions to a new state S_{t+1} and yields a scalar *reward* R_{t+1}. The agent's goal is defined as to maximize the cumulative reward received over time.

In order for the agent to eventually figure out the best available action at every time step, the sequence of states has to provide all relevant information about the environment. Thus, the reinforcement learning problem is formally described as a Markov decision process (MDP), demanding that the sequence of states fulfills the Markov property

$$p(S_{t+1}|S_t) = p(S_{t+1}|S_1, \ldots, S_t) \,, \qquad (2)$$

where $p(S_{t+1}|S_t)$ denotes the conditional probability of transitioning to state S_{t+1} given the previous state S_t. At every time step t, the state S_t is thereby required to provide all relevant information about the transition dynamics of the environment. While many problems can be modeled in this form and the Markov property allows precise theoretical statements, it can be difficult to fulfill this requirement in its most rigorous formulation for practical applications. Nevertheless, recent efforts in reinforcement learning research have proven quite successful [23, 24], and lead to a new wave of attention for the field.

Overall, reinforcement learning represents a general-purpose approach to sequential decision problems, which makes it applicable to a wide range of control tasks.

FEEDBACK DESIGN

In order to apply reinforcement learning methods to control of the micro-bunching instability, we need to define the problem as an MDP. Fortunately, the definition of a Markovian process is straightforward. In case of simulating the longitudinal dynamics via VFP solvers, the starting conditions are given by an initial charge distribution and a set of constant parameters (e.g. machine parameters of the storage ring). The temporal evolution of this charge distribution is then simulated by iteratively solving the VFP equation. At every time step, the calculation of the next step is entirely based on the charge distribution of the preceding time step (neglecting constant parameters). Thus, defining the sequence of longitudinal charge distributions as the state signal

$$S_t \doteq \psi_t(z, E) \qquad (3)$$

yields a Markov process, fully satisfying Eq. (2).

To obtain an MDP, we still need to find an action space providing the agent with the opportunity to influence the micro-bunching dynamics in a meaningful way, and a reward function defining the goal of its task. As our primary interest lies in the emitted CSR power, we define the reward function based on the CSR power time series

$$R_t \doteq R_t(P_{t,\mathrm{CSR}}) \,. \qquad (4)$$

Choosing the reward function is a very crucial part of defining any reinforcement learning problem, as the agent will aim to converge towards the behavior that maximizes the received amount of reward, whether this is the intended solution or not. For the problem at hand, i.e. aiming to stabilize the emitted CSR, the choice can be as simple as

$$R_t \doteq \omega_1 \mu_{t':t} - \omega_2 \sigma_{t':t} \,, \tag{5}$$

where $\mu_{t':t}$ and $\sigma_{t':t}$ denote the normalized mean and standard deviation of the time series $P_{t,\mathrm{CSR}}$ in the interval $[t', t]$, and $\omega_{1,2} > 0$ are simple weighting factors. This definition of reward is expressing our desire of having a CSR power signal of high intensity and low fluctuation, which corresponds to a smooth charge distribution that is not significantly changing in time. Whether this is done in the best possible and most desirable way is unclear and still under investigation. However up to now, Eq. (5) has proven to be a quite reasonable choice.

Finally, we need to define an action space. As the additional CSR wake potential in Eq. (1) acts as a perturbation to the RF potential, one promising approach seems to be centered around the RF system. If we can compensate some of the CSR-driven perturbation, this should have a positive effect on the micro-bunching dynamics. Thus, one straightforward choice of the action space would be

$$A_t \in \{V_{\mathrm{RF}} \times \varphi_{\mathrm{RF}}\} \,, \tag{6}$$

where V_{RF} denotes the RF amplitude and φ_{RF} the RF phase. Dynamically modifying these two parameters should provide the agent with a substantial amount of control over the RF system, however it also includes the option for a trivial solution, as the dependency of the instability threshold on the RF amplitude is well established [5, 25, 26]. Reducing the RF amplitude until the instability threshold is crossed would stabilize the longitudinal dynamics just naturally, but is not what we intend the agent to learn. A slightly modified choice that circumvents this issue is restricting the action space to sinusoidal modulations of the RF amplitude and phase, while maintaining the same effective values

$$A_t \in \{A_V \times f_V \times A_\varphi \times f_\varphi\} \,, \tag{7}$$

where $A_{V,\varphi}$ and $f_{V,\varphi}$ denote the amplitude and frequency of the RF modulations. Based on preliminary studies, the dynamic modulation of the RF amplitude seems to be a particularly suitable and effective choice to counteract the CSR-induced perturbation. The influence of RF modulations on the micro-bunching dynamics has also been tested experimentally in the past, e.g. [27, 28]. A temporally adaptable RF modulation scheme is a promising proposition to exert influence on the longitudinal beam dynamics in the micro-bunching instability as it provides the required flexibility to respond to the varying perturbation by the CSR wake potential over continuous time.

Feasibility of the State Signal

Given the MDP formulation of the problem, as introduced in the previous section, we can apply reinforcement learning methods to solve this task. To do so, the VFP solver Inovesa has already been extended to support dynamic RF modulations and the communication with a reinforcement learning agent during runtime. First tests using this interaction scheme are currently ongoing.

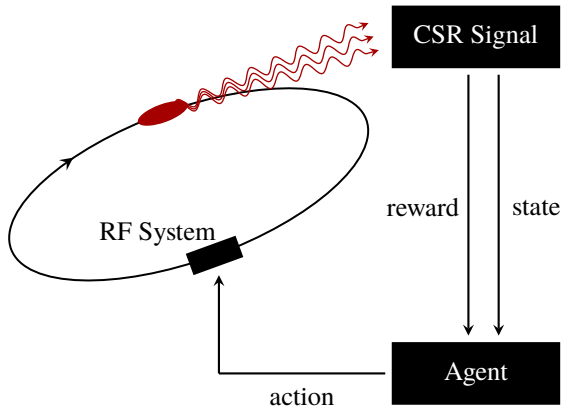

Figure 2: General feedback scheme using the CSR power signal to construct both, the state and reward signal of the Markov decision process (MDP).

The definition of the state signal in Eq. (3) however, is usually not feasible at an actual storage ring. Although first efforts towards phase space tomography have been made at KARA, this type of information is not yet accessible. Instead, we should consider using the diagnostic systems, which are already in place and can provide information about the micro-bunching dynamics. As the projection of the charge distribution $\psi_t(z, E)$ in phase space, the longitudinal bunch profile can be measured using an electro-optical near-field setup on a turn-by-turn basis [29–31]. Complementary information about the second dimension of the longitudinal phase space, i.e. the energy, can be gained by measuring the horizontal bunch profile in a dispersive section of the accelerator using a fast-gated camera [32–34]. However, the simplest and most robust way of acquiring information about the micro-bunching dynamics is by using the CSR power signal $P_{t,\mathrm{CSR}}$ itself. In order to calculate the reward function defined in Eq. (4), we need to measure $P_{t,\mathrm{CSR}}$ regardless of the definition used for the state signal. As the emitted CSR power and its fluctuation over time are strongly correlated to the micro-bunching dynamics within the bunch, we aim to construct a state signal using merely this information

$$S_t \doteq S_t(P_{t,\mathrm{CSR}}) \,. \tag{8}$$

Figure 2 illustrates the resulting feedback scheme.

In order to represent the required data in condensed form, we would like to construct a feature vector that efficiently describes the current state of the micro-bunching dynamics. Some features which are expected to yield characteristic information about that are combined in this exemplary choice

$$S_t \doteq (\mu_{t':t}, \sigma_{t':t}, m_{t':t}, f_{\max}, A_{\max}, \varphi_{\max})^{\mathrm{T}} \,, \tag{9}$$

where $m_{t':t}$ represents a slow trend in the amplitude of the CSR power. The variables $f_{\max}, A_{\max}, \varphi_{\max}$ denote the fre-

quency, amplitude and phase of the main component in the Fourier transform of the time series $P_{t,\mathrm{CSR}}$ in the preceding interval $[t', t]$. The such modified definition of the state signal is quite different from the initial consideration in Eq. (3), which means we no longer have the theoretical comfort of perfectly fulfilling the Markov property. Whether the definition in Eq. (9) yields enough information for the agent to choose adequate actions is unclear and has to be verified in practice. Ideally, this compact definition of the state signal results in a fast learning process and convergence to a satisfying extent of control over the micro-bunching dynamics and thereby the emission of CSR. If these goals can not be met experimentally, the state signal should be extended to carry more information in order to satisfy the Markov property in Eq. (2) as closely as possible.

Finally, we need to consider the step width $\Delta t = t - t'$ within the MDP, which corresponds to the feedback's repetition rate. As the micro-bunching dynamics and the changes caused by the agent's actions occur at time scales governed by the synchrotron period, the step width has to be chosen small enough in order to react to these fast changes. Whether or not this can be relaxed to slower interaction rates has to be tested empirically. At KARA, the synchrotron period is usually in the order of several kHz, which yields challenging time constraints for the hardware implementation of this feedback scheme.

HARDWARE IMPLEMENTATION

The FGPA DAQ Board

To face up to the upcoming demand of high data throughput and fast data processing close to the data source, a novel PCIe readout card is developed at KIT. The DAQ board is shown in Fig. 3.

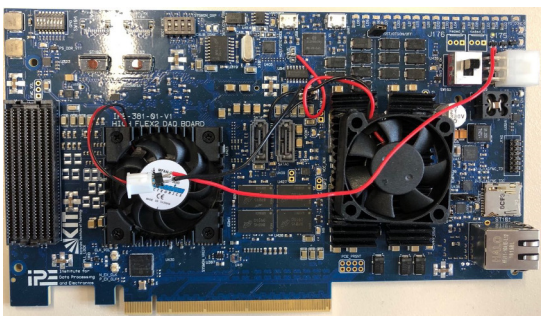

Figure 3: Shown is the novel PCIe ZYNQ MPSoC Data Aquisition Board developed at KIT.

The main processor is based on the ZYNQ UltraScale+ targeting the xcu11eg-1ffvc1760 Xilinx device, which can be divided into two parts: Programmable Logic (PL, FPGA) side and the Processing Subsystem (PS, ARM). It includes a 64-bit quad-core ARM processor with up to 1.5 GHz and a dual-core ARM with up to 600 MHz for real-time tasks. A Mali-400 GPU is available for simple parallel data processing. The ZYNQ is equipped with a large FPGA with about 600k Configurable Logic Blocks (CLB) and several tens of megabytes of block RAM and UltraRAM. The selected FPGA contains more than 2900 DSP slices [35] and can thus fulfill the synthesis and implementation requirement of machine learning.

KAPTURE-2 Front-End Electronic

KAPTURE-2 (Karlsruhe Pulse Taking Ultra-fast Readout Electronics) [36] is a picosecond sampling system for THz pulses at high repetition rates, as produced by synchrotron light sources (2 ns at KARA) due to the high frequency (500 MHz) of the accelerating RF system. KAPTURE-2 is able to acquire and sample the pulse shape with 3 ps resolution by 4 channels simultaneously and continuously, with a data rate $4 \times 1.8\,\mathrm{GS/s}$ at 12 bit per sample point (see Fig. 4).

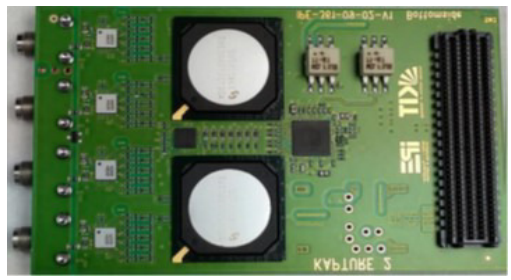

Figure 4: Shown is the 4 channel picosecond sampling system developed at KIT.

Hardware Implementation Scheme

From the hardware point of view, the main difference between supervised and reinforcement learning is that the former can usually be done using an offline training process while the latter requires online training. For reinforcement learning, the learning process is continuous and has to happen during runtime in order to allow the agent to learn from past experience and to efficiently explore its action space. This yields much higher demands regarding the hardware implementation. An iteration of both, the training process and the inference process, need to be completed within the

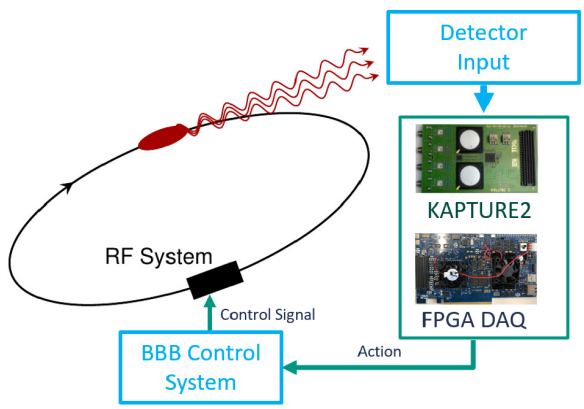

Figure 5: The hardware implementation scheme needs to satisfy the demand for kHz repetition rate.

challenging time constraints specified by the feedback loop illustrated in Fig. 5.

As discussed above, the timing requirements for the feedback repetition rate are resulting directly from the physics time-scale and are in the order of kHz. That means the timing from the detector, the data sampling by the front-end electronics, the FPGA or ARM neural network inference and the control signal generation for the RF system need to finish within 1 ms. The data collection and the training must also be performed in this narrow time window. This requires a special implementation supporting the reinforcement learning approach. The whole training and inference process need to be run directly on the hardware.

Implementation of NN Inference on FPGA

The major task of implementing machine learning (ML) on an FPGA is to transfer the (deep) learning model to the FPGA architecture. This section will demonstrate one solution to map ML models to the ZYNQ UltraScale+.

The implementation tool set for mapping the ML model to FPGA is called HLS4ML [37]. It transfers ML models implemented in the supported ML frameworks, while using a high-level neural networks Python-API as Keras [38] or Pytorch [39], to the altered hardware.

HLS4ML is suitable with all the current Xilinx FPGA devices like the Virtex or Kintex series, because it transfers the model first to the high level synthesis (HLS) project. Afterwards, the HLS code or IP core can be used in the FPGA implementation. The workflow to generate HLS code and the final firmware implementations of machine learning algorithms is shown in Fig. 6.

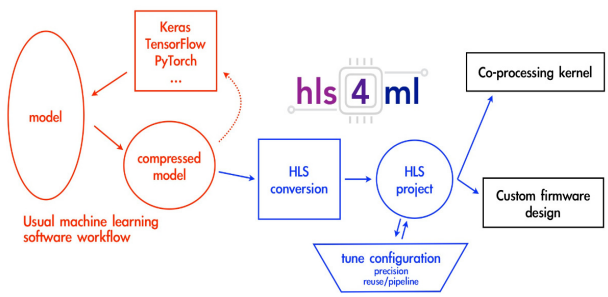

Figure 6: The HLS4ML framework transfers the ML model from Python to an HDL based model.

Full Implementation of Policy Gradient on ARM

Selection of RL environment for testing For a first test of the hardware, a suitable environment needs to be deployed to interact with the hardware as the VFP solver can not be built directly on the hardware platform. Thus, a different RL problem is selected to test the performance of the hardware, which demonstrates the feasibility as the algorithm run on the FPGA or ARM is the same for RF control and for the CartPole control problem [40].

Selection of RL algorithm for testing The performance tests need to include the fast neural network inference which corresponds to the choice of the proper action at the current time step, and the speed of the training process after collecting the states, rewards and actions taken in one episode. Thus, there are three major parts that need to be accomplished on the hardware, a simulation of the environment, the fast inference, and the training process (backward propagation).

The RL method implemented on the FPGA could be any algorithm that can prove the hardware inference and training capability described above. Some examples would be the A3C [41] or the DDPG [42] method.

CartPole problem with policy gradient We will consider solving the CartPole problem specifically by using a simple policy gradient method. The CartPole is fully implemented in C code on the ARM of the MPSoC. As illustrated in Fig. 7, this environment simulates the CartPole on a horizontal axis, where the pole can be moved by applying actions to the cart (NN output is discretized to match the environment). The goal is to keep the pole balanced (the pole stays in a narrow angle range) as long as possible.

Algorithm 1 shows the basic procedure of a policy gradient method and how the agent interacts with the environment. At every episode, in step 8, the agent chooses one action according to the current state (observation). Then the agent applies this action and transitions to the next state (S_{t+1} in step 9). It also collects a scalar reward from the environment. Then at step 11, the agent needs to store the current step, which includes the state, the chosen action and the received reward. After that, the agent checks whether the episode is finished or not. In our case, a fallen pole means that the episode is finished. If a failure of control happens, the episode is finished and the RL agent collects all information related to this episode and calculates the discounted cumulative reward in step 14. This information is then used to update the parameters of the policy in step 15.

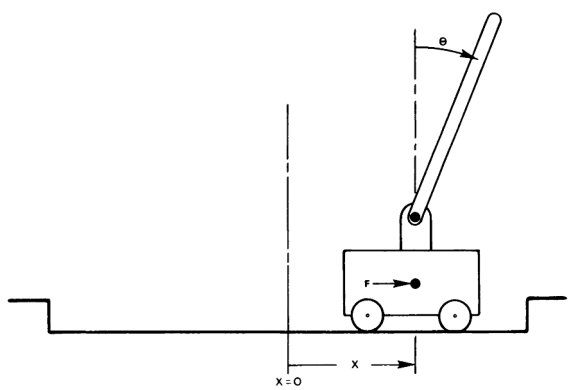

Figure 7: The CartPole environment [40] is used to test the hardware performance on a comparable reinforcement learning problem.

Algorithm 1 Policy Gradient Method (REINFORCE [43])

1: differential policy parameterization $\pi_{\boldsymbol{\theta}}(a|s)$ (fully connected neural network)
2: initialize policy parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ and the learning rate $\alpha > 0$
3: allocate memory to store information about the interaction with the environment
4: **repeat**
5: reset the S_0 to a random starting state
6: $t \leftarrow 0$
7: **repeat**
8: choose action A_t according to $\pi_{\boldsymbol{\theta}}(\cdot|S_t)$
9: $S_t, R_t, S_{t+1} \leftarrow$ env.step(A_t)
10: $t \leftarrow t + 1$
11: store the S_t, A_t, R_t, S_{t+1} in memory
12: **until** S_{t+1} is terminal (state S_T)
13: **loop** over steps in this episode $t = 0, 1, \ldots, T-1$
14: calculate the cumulative discounted reward G_t
15: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha G_t \nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(A_t|S_t)$
16: **end loop**
17: **until** episode has reached a threshold number of steps

If the pole is kept balanced for more than a given number of steps (customized value), the agent is assumed to have learned solving this problem and the training process stops in step 17. In principle, the agent can also be trained forever. In the following, we discuss its implementation on hardware.

CartPole on ZYNQ (ARM) The experiment is done on the processing subsystem side of the ZYNQ. A full reverse engineering on Tensorflow is done for policy gradient and fully implemented on ARM.

In Fig. 8, each blue point represents one episode. The y-axis indicates how many steps the agent achieved in this episode. The threshold for the maximum number of steps (finishing condition) in Algorithm 1 was set to 2000. Thus the training process stops, if the agent manages to keep the pole balanced for more than 2000 steps. As a result, the pole is kept balanced for 2157 steps at the 1663th episode. The entire training process took 161 228.30 µs, 193 472 022 clock cycles. On average, each episode takes 0.096 ms. Due to the usage of a simple policy gradient method, all steps of the episode are considered for backward propagation. While using e.g. the DDPG algorithm instead, updates would be made after every step of the environment, making a single iteration of backward propagation much less expensive.

Compared with the FPGA implementation, this result yields to different options for the implementation: The first option is doing inference on the FPGA, training on ARM and then a parameter assignment from ARM to FPGA. The second option is doing both steps on ARM, being already fast enough.

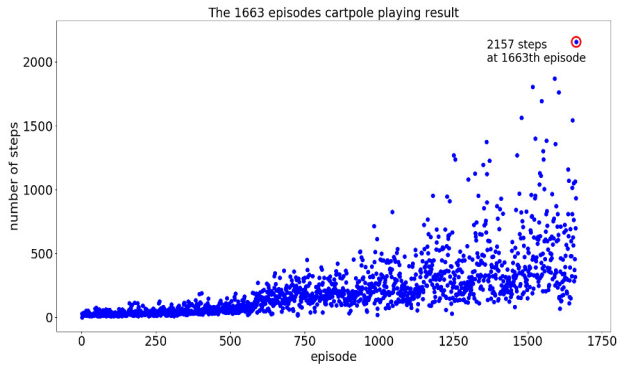

Figure 8: After 1663 episodes in the CartPole environment, the episode threshold is exceeded for the first time.

SUMMARY AND OUTLOOK

Driven by CSR self-interaction, the micro-bunching instability at storage rings is caused by a fast and dynamic perturbation that depends on the longitudinal charge distribution. In order to establish extensive control over these dynamics, we aim for a feedback system that can react to small changes in the charge distribution and adjust the RF system accordingly. As a potent general-purpose approach, reinforcement learning offers the opportunity to model these dynamics and to apply solution methods, which optimize for a pre-defined goal in form of a scalar reward function. The required definition of a Markov decision process is well-motivated due to the inherent Markov property of VFP solvers and conceptually outlined in this contribution. As the micro-bunching dynamics are governed by the synchrotron frequency, the illustrated feedback design yields challenging time constraints for the implementation at the KIT storage ring KARA. Thus, its feasibility is demonstrated on a specialized hardware platform developed at KIT. Both, an FPGA- and an ARM-based implementation are proven to be feasible. Using the textbook CartPole problem as test environment and applying a simple policy gradient method yields results that are comparable to computation on a standard PC, but at dramatically increased in speed for both, training and inference. Beyond the envisaged application, the developed hardware platform can be used for any reinforcement learning task with similar timing requirements. In future work, other RL methods (e.g. DDPG or A3C) and test environments (pendulum, flappy bird) will be considered and tested on the hardware. The task balance between FPGA and ARM is also a promising subject for further investigations. Moreover, the differing case of a fixed neural network implementation on FPGA side will be considered, as this provides relatively low latency and high speed compared to the ARM implementation. Finally, the outlined feedback scheme is also not necessarily restricted to the micro-bunching instability. Different collective effects can be modeled in form of Eq. (1) and simulated using a VFP solver. A successful implementation may thus be easily transferable to control tasks of different longitudinal instabilities at storage rings.

ACKNOWLEDGEMENT

This research is in part supported by the Innovationspool Amalea (Accelerating Machine Learning for Physics) in the Helmholtz Association's Programme "Matter and Technologies". T. Boltz acknowledges the support by the DFG-funded Doctoral School "Karlsruhe School of Elementary Particle and Astroparticle Physics: Science and Technology (KSETA)". W. Wang acknowledges the support by the Major Program of the National Natural Science Foundation of China (no. 61973253) and his supervisor Y. Fang.

REFERENCES

[1] T. Boltz, "Comprehensive Analysis of Micro-Structure Dynamics in Longitudinal Electron Bunch Profiles", Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2017. doi:10.5445/IR/1000068253

[2] T. Boltz *et al.*, "Studies of Longitudinal Dynamics in the Micro-Bunching Instability using Machine Learning", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, May 2018, pp. 3277–3279. doi:10.18429/JACoW-IPAC2018-THPAK030

[3] M. Caselle *et al.*, "Ultrafast linear array detector for real-time imaging", in *Proc. SPIE 10937, Optical Data Science II*, San Francisco, CA, USA, March 2019, pp. 1–9. doi:10.1117/12.2508451

[4] T. Boltz *et al.*, "Feedback Design for Control of the Micro-Bunching Instability based on Reinforcement Learning", in *Proc. 10th Int. Particle Accelerator Conf. (IPAC'19)*, Melbourne, Australia, May 2019, pp. 104–107. doi:10.18429/JACoW-IPAC2019-MOPGW017

[5] K. L. F. Bane, Y. Cai, and G. Stupakov, "Threshold studies of the microwave instability in electron storage rings", *Phys. Rev. ST Accel. Beams*, vol. 13, p. 104402, 2010. doi:10.1103/PhysRevSTAB.13.104402

[6] M. Byrd *et al.*, "Observation of Broadband Self-Amplified Spontaneous Coherent Terahertz Synchrotron Radiation in a Storage Ring", *Phys. Rev. Lett.*, vol. 89, p. 224801, 2002. doi:10.1103/PhysRevLett.89.224801

[7] A.-S. Müller *et al.*, "Accelerator-based sources of infrared and terahertz radiation", *Rev. Accel. Sci. Techol.*, vol. 03, p. 165, 2010. doi:10.1142/9789814340397_0009

[8] M. Abo-Bakr *et al.*, "Steady-State Far-Infrared Coherent Synchrotron Radiation detected at BESSY II", *Phys. Rev. Lett.*, vol. 88, p. 254801, 2002. doi:10.1103/PhysRevLett.88.254801

[9] B. E. Billinghurst *et al.*, "Longitudinal bunch dynamics study with coherent synchrotron radiation", *Phys. Rev. Accel. Beams*, vol. 19, p. 020704, 2016. doi:10.1103/PhysRevAccelBeams.19.020704

[10] W. Shields *et al.*, "Microbunch instability observations from a THz detector at diamond light source", *Journal of Physics: Conference Series*, vol. 357, p. 012037, 2012. doi:10.1088/1742-6596/357/1/012037

[11] E. Karantzoulis *et al.*, "Characterization of coherent THz radiation bursting regime at Elettra", *Infrared Phys. Technol.*, vol. 53, p. 300, 2010. doi:10.1016/j.infrared.2010.04.006

[12] A. Andersson *et al.*, "Coherent synchrotron radiation in the far-infrared from a 1 mm electron bunch", *Optical Engineering*, vol. 39, p. 3099, 2000. doi:10.1117/1.1327498

[13] F. Wang *et al.*, "Coherent THz Synchrotron Radiation from a Storage Ring with High-Frequency RF System", *Phys. Rev. Lett.*, vol. 96, p. 064801, 2006. doi:10.1103/PhysRevLett.96.064801

[14] J. Feikes *et al.*, "Metrology Light Source: The first electron storage ring optimized for generating coherent THz radiation", *Phys. Rev. ST Accel. Beams*, vol. 14, p. 030705, 2011. doi:10.1103/PhysRevSTAB.14.030705

[15] G. L. Carr *et al.*, "Observation of coherent synchrotron radiation from the NSLS VUV ring", *Nucl. Instrum. Methods Phys. Res.*, vol. 463, p. 387, 2001. doi:10.1016/S0168-9002(01)00521-6

[16] C. Evain *et al.*, "Spatio-temporal dynamics of relativistic electron bunches during the micro-bunching instability in storage rings", *Europhys. Lett.*, vol. 98, p. 40006, 2012. doi:10.1209/0295-5075/98/40006

[17] A. R. Hight Walker *et al.*, "New infrared beamline at the NIST SURF II storage ring", in *Proc. SPIE Int. Soc. Opt. Eng. '97*, San Diego, CA, USA, Oct. 1997, vol. 3153. doi:10.1117/12.290261

[18] A. Mochihashi *et al.*, "Observation of THz Synchrotron Radiation Burst in UVSOR-II Electron Storage Ring", in *Proc. 10th European Particle Accelerator Conf. (EPAC'06)*, Edinburgh, UK, Jun. 2006, paper TH-PLS042, pp. 3380–3382.

[19] R. L. Warnock and J. A. Ellison, "A General Method for Propagation of the Phase Space Distribution, with Application to the Sawtooth Instability", *SLAC Technical Report*, No. SLAC-PUB-8404, 2000. doi:10.2172/753322

[20] P. Schönfeldt *et al.*, "Parallelized Vlasov-Fokker-Planck solver for desktop personal computers", *Phys. Rev. Accel. Beams*, vol. 20, p. 030704, 2017. https://github.com/Inovesa/Inovesa

[21] J. L. Steinmann *et al.*, "Continuous bunch-by-bunch spectroscopic investigation of the microbunching instability", *Phys. Rev. Accel. Beams*, vol. 21, p. 110705, 2018. doi:10.1103/PhysRevAccelBeams.21.110705

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 2018.

[23] V. Mnih *et al.*, "Human-level control through deep reinforcement learning", *Nature*, vol. 518, pp. 529–533, 2015. doi:10.1038/nature14236

[24] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, pp. 484–489, 2016. doi:10.1038/nature16961

[25] P. Kuske, "CSR-driven Longitudinal Single Bunch Instability Thresholds", in *Proc. 4th Int. Particle Accelerator Conf. (IPAC'13)*, Shanghai, China, May 2013, paper WEOAB102, pp. 2041–2043.

[26] M. Brosi *et al.*, "Fast mapping of terahertz bursting thresholds and characteristics at synchrotron light sources", *Phys. Rev. Accel. Beams*, vol. 19, p. 110701, 2016. doi:10.1103/PhysRevAccelBeams.19.110701

[27] Y. Shoji and T. Takahashi, "Coherent Synchrotron Radiation Burst from Electron Storage Ring under External RF Modulation", in *Proc. 11th European Particle Accelerator Conf. (EPAC'08)*, Genoa, Italy, Jun. 2008, paper MOPC048, pp. 178–180.

[28] J. L. Steinmann, "Diagnostics of Short Electron Bunches with THz Detectors in Particle Accelerators", Ph.D. thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2019.

[29] N. Hiller *et al.*, "A Setup for Single Shot Electro Optical Bunch Length Measurements at the ANKA Storage Ring", in *Proc. 2nd Int. Particle Accelerator Conf. (IPAC'11)*, San Sebastian, Spain, Sep. 2011, paper TUPC086, pp. 1206–1208.

[30] L. Rota *et al.*, "KALYPSO: A Mfps Linear Array Detector for Visible to NIR Radiation", in *Proc. 5th Int. Beam Instrumentation Conf. (IBIC'16)*, Barcelona, Spain, Sep. 2016, pp. 740–743. doi:10.18429/JACoW-IBIC2016-WEPG46

[31] S. Funkner *et al.*, "High throughput data streaming of individual longitudinal electron bunch profiles", *Phys. Rev. Accel. Beams*, vol. 22, p. 022801, 2019. doi:10.1103/PhysRevAccelBeams.22.022801

[32] P. Schütze *et al.*, "A Fast Gated Intensified Camera Setup for Transversal Beam Diagnostics at the ANKA Storage Ring", in *Proc. 6th Int. Particle Accelerator Conf. (IPAC'15)*, Richmond, VA, USA, May 2015, pp. 872–875. doi:10.18429/JACoW-IPAC2015-MOPHA039

[33] B. Kehrer *et al.*, "Synchronous detection of longitudinal and transverse bunch signals at a storage ring", *Phys. Rev. Accel. Beams*, vol. 21, p. 102803, 2018. doi:10.1103/PhysRevAccelBeams.21.102803

[34] B. Kehrer *et al.*, "Turn-by-Turn Horizontal Bunch Size and Energy Spread Studies at KARA", in *Proc. 10th Int. Particle Accelerator Conf. (IPAC'19)*, Melbourne, Australia, May 2019, pp. 2498–2500. doi:10.18429/JACoW-IPAC2019-WEPGW016

[35] Xilinx, "Zynq ultrascale+ mpsoc product tables and product selection guide", 2016.

[36] M. Caselle, L. A. Perez, M. Balzer, A. Kopmann, L. Rota, M. Weber, M. Brosi, J. Steinmann, E. Bründermann, and A.-S. Müller, "Kapture-2. a picosecond sampling system for individual thz pulses with high repetition rate", *Journal of Instrumentation*, vol. 12, no. 01, p. C01040, 2017. doi:10.1088/1748-0221/12/01/C01040

[37] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, V. Loncar, J. Ngadiuba, M. Pierini, D. Rankin *et al.*, "Fast inference of deep neural networks for real-time particle physics applications", in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2019, pp. 305–305. doi:10.1145/3289602.3293986

[38] F. Chollet *et al.*, "Keras", https://github.com/fchollet/keras, 2015.

[39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch", in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, Oct 2017.

[40] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems", *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 834–846, 1983. doi:10.1109/TSMC.1983.6313077

[41] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning", in *International conference on machine learning*, 2016, pp. 1928–1937.

[42] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning", *arXiv preprint*, 2015. arXiv:1509.02971

[43] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning", *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992. doi:10.1007/BF00992696