ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA021

FREE-ELECTRON LASER OPTIMIZATION WITH REINFORCEMENT LEARNING

N. Bruchon*, G. Fenu, F. A. Pellegrino, E. Salvato, University of Trieste, Trieste, Italy
G. Gaio, M. Lonza, Elettra Sincrotrone Trieste, Trieste, Italy

Abstract

Reinforcement Learning (RL) is one of the most promising techniques in Machine Learning because of its modest computational requirements with respect to other algorithms. RL uses an agent that takes actions within its environment to maximize a reward related to the goal it is designed to achieve. We have recently used RL as a model-free approach to improve the performance of the FERMI Free Electron Laser. A number of machine parameters are adjusted to find the optimum FEL output in terms of intensity and spectral quality. In particular we focus on the problem of the alignment of the seed laser with the electron beam, initially using a simplified model and then applying the developed algorithm on the real machine. This paper reports the results obtained and discusses pros and cons of this approach with plans for future applications.

INTRODUCTION

Free-Electron Lasers (FELs) are complex systems that require continuous effort by experts in order to maintain the high performance that users demand. For seeded FELs, such as FERMI, there are parameters related to the alignment of the seed laser with the electron beam in addition to the large number of electron beam parameters (many dozens at FERMI) [1–4]. Perhaps the most critical parameter is the spatial-temporal overlap between the electron and laser beams in the modulator undulator, the main source of the FEL instability.

The existing feedback systems [5] are able to maintain a steady FEL intensity by controlling the trajectories of the two beams on a shot-to-shot basis. To ease the procedure the electron beam trajectory is kept steady while the position of the seed laser is varied.

Currently, the maintenance of the optimal superimposition of the two beams is carried out by an automatic procedure that is based on the correlation between FEL intensity and the natural jitter in the seed laser parameters [6]. However, this approach cannot be successful if there is insufficient natural jitter: in this case the introduction of artificial noise can help to find the optimal overlap, but FEL performance is affected by the injected noise.

Typical model-free approaches [7] applied in FEL optimization have some intrinsic limitations: they require the availability of the objective function gradient, they are very sensitive to hyper-parameters, and they do not learn from previous experiences. An interesting option to overcome these limitations is given by Machine Learning algorithms, although these approaches can be extremely time consuming.

Nowadays, different optimization techniques are adopted in various FEL facilities [8]. OCELOT [9] has been developed since 2011 at European XFEL and is currently used at the Stanford Linear Accelerator Center (SLAC) [10, 11] and at Deutsches Elektronen-SYnchrotron (DESY) [12]. [13–15] adopt neural networks to model and control particle accelerators. In addition, [16] presents a proof-of-principle of a model-free approach applied at CERN using Deep Q-Learning.

In this paper we present preliminary results obtained using a simple RL algorithm, Q-Learning with Linear Function Approximation, on FERMI. The experiments have been carried out on two different systems.

ENVIRONMENTS

In this work, two different tasks have been considered. Both of them concern the trajectory control of a laser. The first task uses the service laser in the Electro-Optical Sampling station [17–19], while in the second task the seed laser of the undulator modulator of FERMI Free-Electron Laser is used.

In the EOS station the laser movement system is a standard optical alignment scheme, as shown in Fig. 1. It is composed of two planar tip-tilt mirrors [20], each axis of which is driven by a piezo-motor (horizontal and vertical motors), and two virtual screens based on Charged-Coupled Devices (CCDs) [21]. The position of the laser on the two screens is adjusted by moving the tip-tilts. The goal of the task is to align the laser such that it passes through a pair of predefined Regions of Interest (ROI), that must contain a certain fraction of the laser spot to successfully end an episode of the task. The performance of the agent on the task is measured online by the computing the product of the intensity measured in each of the the ROIs.

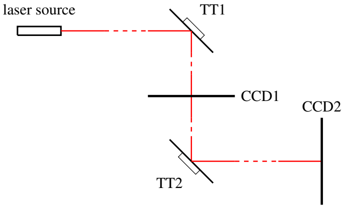


Figure 1: Experimental setup of the EOS laser alignment task. TT1 and TT2 are the tip-tilt mirrors while CCD1 and CCD2 are the virtual screen CCDs.

A simplified representation of the second task, alignment between the seed laser and electron beam at FERMI is shown

[*] niky.bruchon@phd.units.it

in Fig. 2. As in the EOS task the laser trajectory can be moved by the motorized mirrors. Unlike the EOS system the goal is to maximize the value acquired by the intensity monitor (I_0) that measures the energy output of the FEL process.

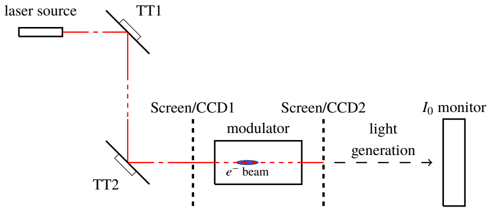


Figure 2: Scheme of the FERMI FEL seed laser alignment set up. TT1 and TT2 are the tip-tilt mirrors, Screen/CCD1 and Screen/CCD2 are the two removable screens with CCDs and I_0 monitor is the employed intensity sensor.

The state $x(t)$, in both systems, is given by sampling at each instant t the voltage applied to the tip-tilts. Since one tip-tilt can be controlled by two voltages (one to move it horizontally and the other to move it vertically) the resulting state, considering both tip-tilts, is a four dimension vector.

REINFORCEMENT LEARNING

Reinforcement Learning (RL) [22, 23] is a sub-field of Machine Learning in which an *agent* learns a *policy* $\pi(u|x)$ by interacting with an environment and maximizing the *rewards* it receives for desired behavior. The basic elements of RL are states $x \in X$ (a measure of the environment), actions $u \in U$ (things the agent does in the environment) and rewards $r(x, u)$ (a scalar function that the agent attempts to maximize). We define u_j as the j-th action. The task is to find the optimal policy $\pi*$ with respect to the maximization of the discounted reward:

$$\sum_{t=0}^{\infty} \gamma^t r(x, u)$$

where $\gamma \in [0, 1]$ is the discount factor.

Q-learning [24] is an algorithm that follows the dynamic programming approach where the *action-value function* $Q(x, u)$ is estimated iteratively. The optimal policy is:

$$\pi^*(u|x) = \arg\max_u Q(x, u)$$

The choice of Q-learning is due to its simplicity and that, in combination with reward shaping [25], it learns efficiently despite sparse rewards.

During learning the exploration is driven by an ϵ-greedy policy, where ϵ is the parameter that defines the probability of a random action (exploration) instead of the best action (exploitation). Furthermore, the update rule is:

$$Q(u, x) \leftarrow Q(u, x) + \alpha\delta$$

where α is the learning rate [26] and δ is the *temporal difference error* (see Algorithm 1 for more details).

The version of Q-learning we adopted works with a continuous state space. This is due to the linear function approximation of the action-value function:

$$Q(x, u_j) = \theta_j^T \phi(x)$$

where $\phi(x)$ is a vector of features and θ_j a vector of weights associated with the action u_j. Hereafter, the corresponding policy will be identified by π_θ.

Many possible choices of linear function approximation are available, we chose Gaussian *Radial Basis Functions* (RBFs):

$$\phi_i(x) = exp\left(-\frac{\| x - c_i \|^2}{2\sigma_i^2}\right)$$

where c_i is a set of centers and σ_i determines the decay rate of the RBF.

The pseudo code of the algorithm used is reported in Algorithm 1.

Algorithm 1 Q-learning algorithm with linear function approximation [27]

```
Initialize $\theta$ and set $\alpha, \gamma$
For each episode:
    Initialize $x$
    Until $x'$ is terminal:
        Choose and perform $u_j \in \mathcal{U}$ using $\pi_\theta$
        Evaluate $x'$ and $r(x, u_j)$

        $\delta \leftarrow r(x, u_j) + \gamma \max_{u_{j'} \in \mathcal{U}} \theta_{j'}^T \varphi(x', u_{j'}) - \theta_j^T \varphi(x, u_j)$

        $\theta \leftarrow \theta + \alpha\delta\varphi(x, u_j)$
        $x \leftarrow x'$
```

IMPLEMENTATION AND RESULTS

The tasks are divided into two parts, an initial training phase followed by a test phase.

As previously described, the state is a four dimensional vector containing the voltages applied to the tip-tilts. Similarly, the input u is a four-element vector indicating the displacement that leads to the next state x'. We worked with a discrete action space and, for this reason, the allowed values of u are fixed to change the voltage applied to the each motor by a given magnitude either positive or negative.

The target value is defined as I_T and it depends on the system: in the EOS one it is given by the product of the intensities detected by the ROIs when the laser spot is centered in each ROI, while in the FEL it is the value detected by the I_0 monitor. At each time step, the input selected by the controller is applied and the new intensity $I_D(x')$ is compared with the I_T. The episode ends in two cases:

- In the new state a certain percentage of the target intensity is reached, i.e. $I_D(x') \geq \% \times I_T$, i.e. the goal is achieved.
- The number of steps reaches an upper limit without reaching the goal.

During training the ϵ and α values decay following the rules derived from [28, 29]:

$$\alpha \leftarrow \alpha \cdot \frac{N_0 + 1}{N_0 + \#episode}, \quad \epsilon \leftarrow \frac{1}{\#episode}; \qquad (1)$$

where the N_0 value is set empirically. Furthermore, the reward is shaped according to [25]:

$$r(x, u) = R + k \cdot \frac{\gamma_{rs}\, I_D(x') - I_D(x)}{I_T}, \qquad (2)$$

where R is taken equal to 1 if the target is reached, 0 otherwise; while the γ_{rs} and k values are set empirically. During the test phase ϵ and α are kept fixed, following [30]. The values of these parameters are presented in Table 1 for both tasks.

At the end of each episode a new episode starts from a randomly selected initial state until the maximum number of episodes is reached. When all the training episodes are completed, the test phase begins while maintaining the target conditions previously defined. Once again, each episode begins with the a randomly selected initial state. The values of the hyper-parameters used for both systems are listed in Table 1.

Table 1: RL General Hyper-parameters

Parameters	**Treaining**	**Test**
max num. of steps	10000	10000
σ^2_{RBF}	0.0075	0.0075
initial ϵ	1	0.05
initial α	0.1	-
N_0 in α decay	20	-
γ	0.99	-
γ_{rs}	0.99	-
k	1	-
training episodes	300	300
test episodes	100	50
motor step size (a.u.)	3000	1000

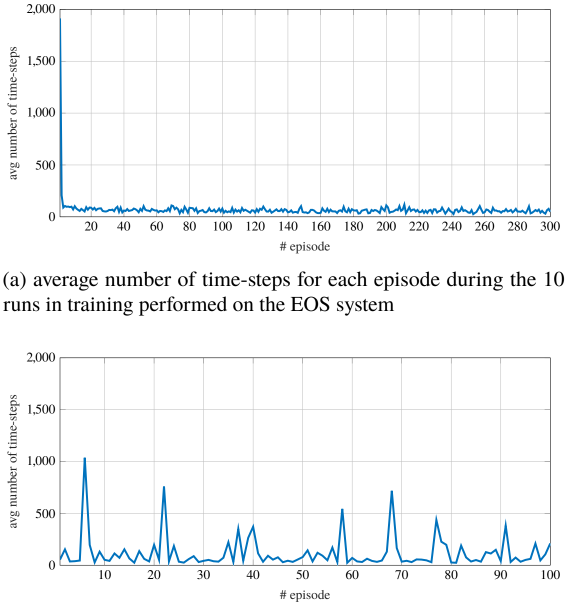
The remaining task parameters are the following:

- % of I_T: this percentage on EOS goes from 95% in training to 90% in test, while on FEL it is set to 92.5% in training and 90% in test. If the agent reaches these values in the respective tasks, the episode is considered successfully completed.
- The full scale range for the voltage applied to the motors spans from 0 to 262144 arbitrary units.

The averaged results obtained during experiments on EOS are presented in Fig. 3.

The implementation of the algorithm on FERMI FEL has been done only once due to the high request of the light source. However, the results obtained during this single run are shown in Fig. 4 and seem to be promising.

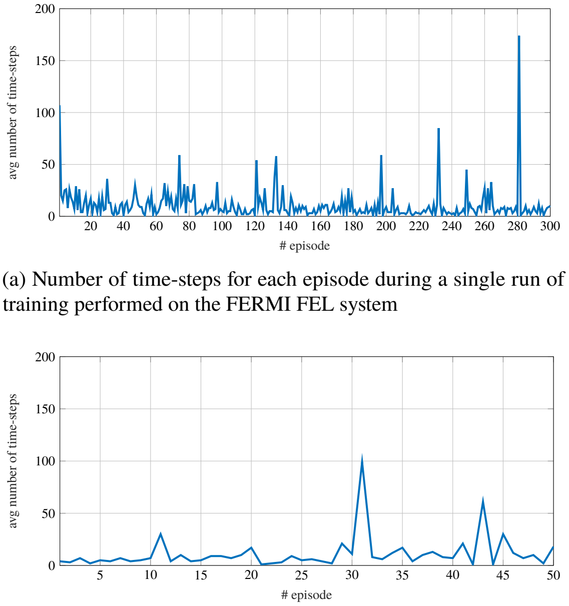
The training plots of both systems show the effectiveness of reward shaping [25]. In both cases the exploration that occurs in the first episode is sufficient for the next episodes to solve the problem in a considerably smaller number of time-steps. However, a new hyper-parameter, k has been introduced by the reward shaping[31]. This value must be empirically tuned to ensure a good balance between the reward R at the end of the successful episodes and the shaping contribution. Furthermore, in the episodes in the test phase that show a peak in the number of steps, the agent placed the beam very close to the target without reaching it. Our conjecture is that this is a consequence of an improper set-



(a) average number of time-steps for each episode during the 10 runs in training performed on the EOS system

(b) average number of time-steps for each episode during the 10 runs in test performed on the EOS system

Figure 3: EOS system results.



(a) Number of time-steps for each episode during a single run of training performed on the FERMI FEL system

(b) number of time-steps for each episode during a single run of test performed on the FERMI FEL system

Figure 4: FEL system results.

ting of the pair k, R during training. Further analysis will be carried out in future works.

CONCLUSIONS

A model free-approach Reinforcement Learning has been implemented on two different systems at Elettra Sincrotrone Trieste. The Q-learning algorithm was able to control the laser alignment process by learning the correct state-action association with only knowledge of the laser beam intensity detected. The positive results obtained from both tasks with this preliminary work has motivated further work on Reinforcement Learning control at the FERMI FEL facility.

Future work will regard the evaluation of a deep Q-learning algorithm [30] first and than the introduction of other RL methods and techniques (e.g. the actor-critic algorithm [32]), to move towards automatic optimization of the FEL facility.

REFERENCES

[1] L. H. Yu, “Generation of intense UV radiation by subharmonically seeded single-pass free-electron lasers,” *Phys. Rev. A*, vol. 44, no. 8, p. 5178, 1991. doi:10.1103/PhysRevA.44.5178

[2] E. Allaria *et al.*, “The FERMI free-electron lasers,” *J. Synchrotron Radiat.*, vol. 22, no. 3, pp. 485–491, 2015. doi:10.1107/S1600577515005366

[3] E. Allaria *et al.*, “Highly coherent and stable pulses from the FERMI seeded free-electron laser in the extreme ultraviolet,” *Nat. Photonics*, vol. 6, no. 10, pp. 699–704, 2012. doi:10.1038/nphoton.2012.233

[4] E. Allaria *et al.*, “Two-stage seeded soft-x-ray free-electron laser,” *Nat. Photonics*, vol. 7, no. 11, pp. 913–918, 2013. doi:10.1038/nphoton.2013.277

[5] G. Gaio and M. Lonza, “Evolution of the FERMI beam based feedbacks,” in *Proc. ICALEPCS’13*, paper THPPC129, San Francisco, CA, USA, 2013, pp. 1362–1365.

[6] G. Gaio, M. Lonza, N. Bruchon, and L. Saule, “Advances in automatic performance optimization at FERMI,” in *Proc. ICALEPCS’17*, Barcelona, Spain, 2017, pp. 352–356. doi:10.18429/JACoW-ICALEPCS2017-TUMPA07

[7] N. Bruchon, G. Fenu, G. Gaio, M. Lonza, F. A. Pellegrino, and L. Saule, “Free-electron laser spectrum evaluation and automatic optimization,” *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 871, pp. 20–29, 2017. doi:10.1016/j.nima.2017.07.048

[8] S. Tomin *et al.*, “Progress in automatic software-based optimization of accelerator performance,” in *Proc. IPAC’16*, Busan, Korea, 2016, pp. 3064–3066. doi:10.18429/JACoW-IPAC2016-WEPOY036

[9] I. Agapov, G. Geloni, S. Tomin, and I. Zagorodnov, “OCELOT: A software framework for synchrotron light source and FEL studies,” *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 768, pp. 151–156, 2014. doi:10.1016/j.nima.2014.09.057

[10] M. McIntire, T. Cope, D. Ratner, and S. Ermon, “Bayesian optimization of FEL performance at LCLS,” in *Proc. IPAC’16*, Busan, Korea, 2016, pp. 3064–3066. doi:10.18429/JACoW-IPAC2016-WEPOW055

[11] M. McIntire, D. Ratner, and S. Ermon, “Sparse gaussian processes for bayesian optimization,” in *Proc. UAI’16*, Jersey City, NJ, USA: AUAI Press, 2016, pp. 517–526.

[12] I. Agapov, G. Geloni, and I. Zagorodnov, “Statistical optimization of FEL performance,” in *Proc. IPAC’15*, Richmond, VA, USA, 2015, pp. 1496–1498. doi:10.18429/JACoW-IPAC2015-TUPWA037

[13] A. L. Edelen, J. P. Edelen, S. G. Biedron, S. V. Milton, and P. J. van der Slot, “Using neural network control policies for rapid switching between beam parameters in a free-electron laser,” *NIPS 2017*, 2017.

[14] A. L. Edelen, S. V. Milton, S. G. Biedron, J. P. Edelen, and P. J. M. van der Slot, “Using a neural network control policy for rapid switching between beam parameters in an FEL,” in *Proc. FEL’17*, Santa Fe, NM, USA, 2017, pp. 480–483. doi:10.18429/JACoW-FEL2017-WEP031

[15] A. Edelen, S. Biedron, B. Chase, D. Edstrom, S. Milton, and P. Stabile, “Neural networks for modeling and control of particle accelerators,” *IEEE Trans. Nucl. Sci.*, vol. 63, no. 2, pp. 878–897, 2016. doi:10.1109/TNS.2016.2543203

[16] S. Hirlaender, V. Kain, and M. Schenk, “New paradigms for tuning accelerators: Automatic performance optimization and first steps towards reinforcement learning at the CERN low energy ion ring,” in *2nd ICFA Workshop on Machine Learning for Charged Particle Accelerators*, Villigen, Switzerland, 2019. https://indico.cern.ch/event/784769/contributions/3265006/attachments/1807476/2950489/CO-technical-meeting-_Hirlaender.pdf

[17] M. Veronese *et al.*, “New results of FERMI FEL1 EOS diagnostics with full optical synchronization,” in *Proc. IBIC’14*, paper MOPD10, Monterey, CA, USA, 2014, pp. 165–168.

[18] M. Veronese, M. Danailov, and M. Ferianis, “The electro-optic sampling stations for FERMI@ Elettra, a design study,” in *Proc. BIW’08*, paper TUPTPF026, Tahoe City, CA, USA, 2008, pp. 158–161.

[19] M. Veronese *et al.*, “First operation of the electro optical sampling diagnostics of the FERMI@ Elettra FEL,” in *Proc. IBIC’12*, paper TUPA43, Tsukuba, Japan, 2012, pp. 449–452.

[20] S. Cleva, L. Pivetta, and P. Sigalotti, “Beaglebone for embedded control system applications,” in *Proc. ICALEPCS’13*, paper MOMIB05, Tsukuba, Japan, 2013, pp. 62–65.

[21] G. Gaio and M. Lonza, “Automatic fel optimization at FERMI,” in *Proc. ICALEPCS’15*, paper MOC3O03, Melbourne, Australia, 2015, pp. 26–29.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[23] B. Recht, “A tour of reinforcement learning: The view from continuous control,” *Annu. Rev. Control Rob. Auton. Syst.*, vol. 2, pp. 253–279, 2018. doi:10.1146/annurev-control-053018-023825

[24] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992. doi:10.1007/BF00992698

[25] A. Y. Ng, D. Harada, and S. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *Proc. ICML’99*, Morgan Kaufmann Publishers Inc., 1999, pp. 278–287.

[26] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951. https://www.jstor.org/stable/

2236626

[27] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 4, no. 1, pp. 1–103, 2010. doi:10.2200/S00268ED1V01Y201005AIM009

[28] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, J. P. How, *et al.*, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 6, no. 4, pp. 375–451, 2013. doi:10.1561/2200000042

[29] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Machine Learning: ECML 2005*, J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 437–448, ISBN: 978-3-540-31692-3.

[30] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 529–533, no. 7540, p. 529, 2015. doi:10.1038/nature14236

[31] M. J. Mataric, "Reward functions for accelerated learning," in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 181–189. doi:10.1016/B978-1-55860-335-6.50030-1

[32] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems 12*, 2000, pp. 1008–1014. http://papers.nips.cc/paper/1786-actor-critic-algorithms.pdf