
Autonomous Pressure Control in MuVacAS via Deep Reinforcement Learning and Deep Learning Surrogate Models

Guillermo Rodriguez-Llorente*

HI Iberia
Juan Hurtado de Mendoza 14
28036 Madrid, Spain.
grodriguez@hi-iberia.es
Department of Mathematics,
Universidad Carlos III de Madrid,
Gregorio Millán Barbany Institute,
Universidad Carlos III de Madrid
28911 Leganés, Madrid, Spain
guirodri@inst.uc3m.es

Galo Gallardo Romero

HI Iberia
Juan Hurtado de Mendoza 14
28036 Madrid, Spain
ggallardo@hi-iberia.es

Rodrigo Morant Navascués

HI Iberia
Juan Hurtado de Mendoza 14
28036 Madrid, Spain
rmorant@hi-iberia.es

Nikita Khvatkin Petrovsky

HI Iberia
Juan Hurtado de Mendoza 14
28036 Madrid, Spain
nkhvatkin@hi-iberia.es

Anderson Sabogal

IFMIF-DONES Spain
18130 Escúzar, Granada Spain
anderson.sabogal@ifmif-dones.es

Roberto Gómez-Espinosa Martín*

HI Iberia
Juan Hurtado de Mendoza 14
28036 Madrid, Spain
robertogemartin@hi-iberia.es

Abstract

The development of nuclear fusion requires materials that can withstand extreme conditions. The IFMIF-DONES facility, a high-power particle accelerator, is being designed to qualify these materials. A critical testbed for its development is the MuVacAS prototype, which replicates the final segment of the accelerator beamline. Precise regulation of argon gas pressure within its ultra-high vacuum chamber is vital for this task. This work presents a fully data-driven approach for autonomous pressure control. A Deep Learning Surrogate Model, trained on real operational data, emulates the dynamics of the argon injection system. This high-fidelity digital twin then serves as a fast-simulation environment to train a Deep Reinforcement Learning agent. The results demonstrate that the agent successfully learns a control policy that maintains gas pressure within strict operational limits despite dynamic disturbances. This approach marks a significant step toward the intelligent, autonomous control systems required for the demanding next-generation particle accelerator facilities.

*Corresponding author.

1 Introduction

Making commercial nuclear fusion energy viable remains a formidable scientific and engineering challenge. A critical step toward this goal is the development of materials capable of withstanding the extreme irradiation conditions inside a reactor. To bridge the gap between current materials science and the requirements of future fusion power plants, the International Fusion Materials Irradiation Facility – Demo Oriented NEutron Source (IFMIF-DONES) is under construction. This unique particle-accelerator-based facility will produce an intense neutron flux by colliding a high-energy deuteron beam with a liquid lithium target, thereby enabling the qualification and testing of candidate materials under prototypical fusion conditions [Bernardi et al., 2022]. To support the safe and reliable design of IFMIF-DONES, the Multipurpose Vacuum Accident Scenarios (MuVacAS) prototype was developed. MuVacAS serves as a dedicated testbed that physically replicates the final segment of the IFMIF-DONES accelerator beamline, where the deuteron–lithium stripping reaction occurs [Sabogal et al., 2023].

A critical aspect of the MuVacAS operation is the precise regulation of pressure within its ultra-high vacuum system. This is achieved through the controlled injection of argon gas to maintain the specific pressure levels required for the stripping reaction, a task complicated by the system’s non-linear dynamics and need to respond to fast, unpredictable disturbances. Traditional control methodologies, such as PID controllers, typically require conservative tuning. While Deep Reinforcement Learning (DRL) is well-suited for such problems [Degraeve et al., 2022], its direct application is slow and risky for the physical equipment. To overcome this barrier, this work presents a data-driven approach: first, a Deep Learning Surrogate Model (DLSM) is trained on real data from the machine to create a safe fast-simulating environment. This environment is then used to train a DRL agent to learn pressure regulation under dynamic disturbances. Although the joint use of these two types of models has been explored in prior studies [Rodríguez-Llorente et al., 2024, Wang et al., 2021], such approaches remain rare and have so far been limited to simulation-only contexts. The results from this work demonstrate that the agent trained in this way learns a robust policy that maintains pressure within strict limits, outperforming conventional methods and marking a significant step towards the intelligent, autonomous control required for next-generation scientific facilities.

2 Methodology

2.1 Data acquisition

The dataset for this study comprises approximately nine hours of real operational data acquired from the MuVacAS prototype. Measurements were sampled at approximately 1 Hz and include synchronized records of the argon injection rate together with pressure readings from a network of 12 cold-cathode sensors distributed along the vacuum line, from the beam entrance at the beginning of the prototype to the collision chamber at the end, where the deuteron–lithium reaction takes place. Sensors 1 and 2, located near the beam entrance, were excluded from the analysis due to their placement in a low-pressure, high-noise region. The argon injection, positioned near the collision chamber, was dynamically modulated using a random Gaussian field technique (with smoothing parameter α of 2.5 and 3) to create diverse yet smoothly varying training conditions [Lang and Potthoff, 2011]. These profiles, with injection rates ranging from 0 to 1 sccm (standard cubic centimeters per minute), were executed automatically on the physical system via a custom script interfacing with the EPICS control system [Dalesio et al., 1993]. The resulting pressure measurements, spanning several orders of magnitude from approximately 10^{-6} mbar to 10^{-3} mbar across the sensor array, were log-transformed for model stability.

2.2 Deep Learning Surrogate Model

The selection of an appropriate surrogate model architecture is critical for creating a high-fidelity digital environment. For this task, the Fourier Neural Operator (FNO) was selected. Unlike traditional neural networks that map between Euclidean spaces, this architecture is specifically designed to learn mappings between infinite-dimensional function spaces. The FNO performs convolution in the frequency domain via the Fast Fourier Transform (FFT), enabling it to efficiently capture long-range spatial dependencies and global correlations [Li et al., 2020]. Formally, this model is trained with the dataset created in Section 2.1 to approximate the gas pressure evolution in time along the accelerator

longitude z , as described by the following operator:

$$q_t, p_t(z) \xrightarrow{\text{FNO}} p_{t+1}(z),$$

Where at a given time t , the model takes the argon injection rate q_t and the entire spatial pressure distribution $p_t(z)$, and predicts the resulting pressure distribution at the next time step, $p_{t+1}(z)$. This single-step, Markovian formulation is a direct requirement for its integration within the DRL training loop.

The model was implemented and trained using the NVIDIA Modulus framework² on a final dataset of approximately 30,000 instances (environment transitions with argon injections). The dataset was partitioned into 80% for training and 20% for testing, and the model’s weights were optimized using the Adam algorithm to minimize the mean squared error between its predictions and the ground-truth pressure distributions from the MuVacAS prototype. See Appendix B for more details on the model hyperparameters. Finally, an iterative evaluation of the model given an initial pressure distribution and a set of argon injections through time is displayed in Appendix A.

2.3 Deep Reinforcement Learning

To autonomously regulate the pressure at the lithium target (monitored by the sensor located at the end of the beamline), a DRL agent was trained within a simulated environment constructed around the FNO surrogate model (Section 2.2) and implemented using the Gymnasium framework [Towers et al., 2024]. The agent’s objective is to learn an optimal policy function π that adjusts the argon injection at each time step in order to reach the desired pressure in the collision chamber as rapidly as possible and maintain it thereafter. For training, the selected algorithm was Proximal Policy Optimization (PPO) [Schulman et al., 2017], already implemented in the Stable-Baselines3 (SB3) library [Raffin et al., 2021]. PPO is an on-policy, gradient-based method that has become a standard in continuous control tasks due to its training stability, natural handling of continuous action spaces and wide adoption and proved performance across multiple works. In this algorithm, the policy, i.e., the decision-making mechanism of the agent, is represented by a neural network (see Figure 3 in Appendix C). At each step, the agent outputs a single continuous action corresponding to the argon mass flow rate, which directly influences the pressure dynamics.

The observation space provided to the agent consists of 3 values. It contains the log-transformed pressure readings from the sensor located at the collision chamber, the log-transformed pressure objective and the current injection rate. The action space is simply a scalar to control the argon injection rate. The ranges of all variables are given by the values chosen in the dataset creation stage in Section 2.1 and can be seen in detail in Table 2 of Appendix C. The agent’s behaviour is guided by a single-term exponential reward function that depends only on the absolute difference between the current and target pressures (evaluated in the log-pressure space). Concretely, at time step t the reward is:

$$r_t = \exp(-k |p_t - p^*|),$$

where p_t is the current (log) pressure at the collision chamber, p^* is the desired objective (log) pressure, and $k = 4$ is a sharpness coefficient controlling how quickly the reward decays with error. This function attains its maximum value $r_t = 1$ when $p_t = p^*$ and decreases smoothly toward 0 as the deviation grows; it therefore provides stable, non-negative, and well-conditioned gradients for training. Further details on the agent parameters can be seen in Table 3 of Appendix C.

3 Results

First, the trained DRL agent was systematically evaluated in simulated environments with dynamically assigned pressure targets that were not included in the training set. The corresponding results are presented in Appendix D. In these evaluations, the agent successfully regulated the argon injection to reach and maintain multiple target pressures. An interesting observation is the asymmetry between upward and downward transitions: achieving lower pressure states generally required more control actions than increasing the pressure, which is consistent with the underlying physical dynamics of the system.

²Now known as NVIDIA Physics NeMo: <https://developer.nvidia.com/physicsnemo>

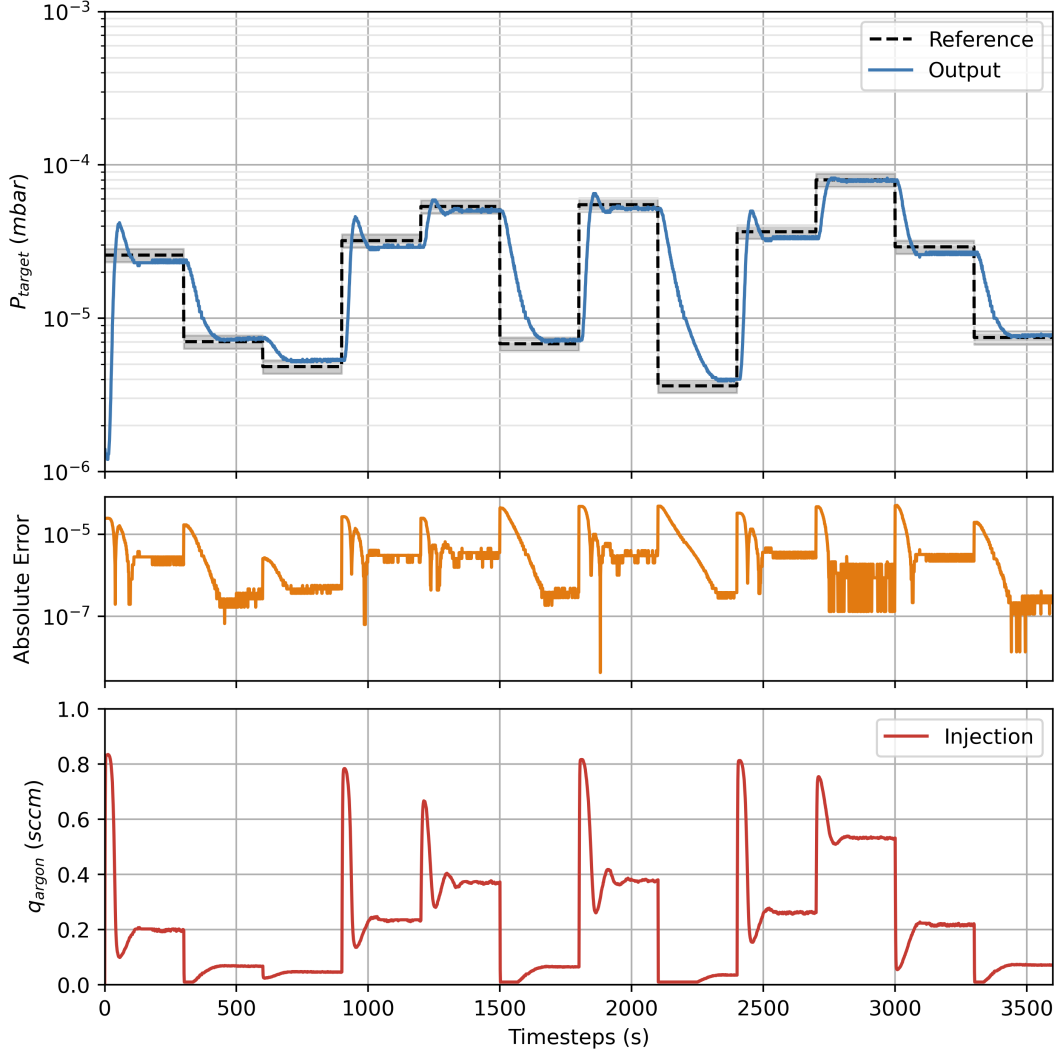


Figure 1: Evaluation of the DRL agent on the real prototype. Top: achieved and reference pressures. Middle: absolute error between the achieved and reference pressures. Bottom: argon injection values. Here p_{target} denotes the pressure in the lattice section where the collision chamber is located.

Following the positive results of the validation in the simulated environment, the agent was deployed in the physical MuVacAS prototype. A sequence of progressively complex experiments was conducted, ranging from maintaining fixed pressure targets to tracking dynamically varying setpoints over extended periods. During deployment, built-in safety protocols were active, including automatic shutdown upon exceeding pressure limits and the ability for a human operator to assume manual control. All agent actions and observations were logged in real time, providing a comprehensive dataset for evaluating the performance and stability of the autonomous control policy under real-world conditions (Figure 1 and Appendix E). Notably, due to time constraints, the experimental setup featured a different argon injection location than the one encountered during training. Despite this change, the agent successfully reached and maintained all desired pressure configurations, though it required a larger number of control actions to do so.

4 Conclusions and future work

The deployment of the DRL agent on the physical MuVacAS prototype marks a significant step toward autonomous control of complex, non-linear subsystems in advanced scientific facilities. Trained on data from a prototype configuration in which argon injection occurred at the collision chamber, the agent was later deployed with the injector positioned further away (consistent with the geometry foreseen for the final accelerator design) to reduce irradiation exposure. This modification altered the pressure dynamics and introduced a distribution shift between the training and deployment conditions. Despite this, the agent successfully regulated argon injection to reach and maintain target pressures within operational bounds, even under dynamic scenarios not encountered during training. In contrast, proportional–integral–derivative (PID) controllers tuned under the previous configuration were unable to achieve satisfactory performance. The DRL agent’s adaptive and robust behaviour thus validates the feasibility of the learned policy and demonstrates the superior generalization capabilities of DRL compared with traditional control strategies. Broadly, these results provide a compelling proof-of-concept for AI-driven regulation of critical vacuum and gas injection systems in nuclear fusion facilities such as IFMIF-DONES, highlighting the potential of DRL to manage complex systems while reducing the operational load on human operators.

A key enabling factor in this success is the synergy between DLSMs and DRL. By providing a high-fidelity yet computationally efficient digital twin of the argon injection system, the FNO-based surrogate allowed the agent to be trained orders of magnitude faster than would be feasible on the physical system or with conventional simulators (e.g., Molflow [Kersevan and Ady, 2019]), where each control step requires approximately one second. This underscores the critical role of DLSMs in making DRL practical for real-world scientific applications. Beyond serving as safe and accelerated training environments, DLSMs can be used to construct digital twins of accelerator subsystems thanks to their near real-time inference capability, and they further open the door to advanced optimization strategies such as gradient-based methods due to their differentiability [Gallardo et al., 2025].

Although the main objective of this study was to demonstrate the synergy between DLSMs and DRL for autonomous pressure control, it is acknowledged that alternative learning paradigms could also address this problem effectively given the available data. Offline reinforcement learning methods, which derive optimal control policies directly from pre-collected datasets without further interaction with the environment, and imitation learning approaches, which learn control strategies from expert demonstrations, represent promising complementary directions. Future work will explore the implementation and benchmarking of these techniques against the current DLSM–DRL framework.

Future iterations of the system will also aim to restrict the agent’s observation space to pressure readings obtained exclusively outside the collision chamber, thereby improving its suitability for deployment in the final accelerator configuration. In parallel, several additional enhancements are foreseen to further improve the agent’s performance. Fine-tuning the pre-trained policy on the real machine could help close the remaining sim-to-real gap [Zhao et al., 2020] and enhance robustness, particularly in regimes the agent has not previously encountered, such as modified injection setups. Moreover, the surrogate model’s accuracy can be improved by incorporating more representative transition states, as Gaussian random fields do not capture static or slowly varying states. Finally, planned future experiments will include the original injection to benchmark the agent against traditional controllers performance.

Acknowledgments and Disclosure of Funding

We wish to acknowledge and thank the other participants of DONES-FLUX project, the IFMIF-DONES consortium as well as the University of Granada for their role played in the development of this work. The project DONES-FLUX, with file number MIP-20221017, has been subsidised by the CDTI—Centro de Desarrollo Tecnológico Industrial through the call for proposals of the “Misiones CDTI” programme for the year 2022 and it is supported by the Ministry of Science and Innovation of Spain.

References

- Davide Bernardi, Angel Ibarra, Frederik Arbeiter, F. Arranz, Mauro Cappelli, Philippe Cara, J. Castellanos, Hervé Dzitko, A. García, Jandri Gutiérrez, W. Królas, Francisco Martín-Fuertes, G. Micciché, Antonio Muñoz Roldan, F. Nitti, Tonio Pinna, Ivan Podadera, J. Pons, Yuefeng Qiu, and Raquel Román. The ifmif-dones project: Design status and main achievements within the eurofusion fp8 work programme. *Journal of Fusion Energy*, 41:24, 10 2022. doi: 10.1007/s10894-022-00337-5.
- L. R. Dalesio, J. O. Hill, M. Kraimer, S. Lewis, D. Murray, S. Hunt, M. Claussen, W. Watson, and J. Dalesio. The experimental physics and industrial control system architecture: Past, present, and future. In *Proceedings of the International Conference on Accelerators and Large Experimental Physics Control Systems (ICALEPCS '93)*, Berlin, Germany, Nov 1993. Los Alamos National Laboratory. URL <https://www.osti.gov/biblio/10193541>.
- Jonas Degraeve, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristiano Sommariva, Stefano Coda, Bruno Lepape, Martin Riedmiller, Olivier Bachem, Rupesh Srivastava, Timothy P. Lillicrap, and Martin A. Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, February 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04301-9. URL <https://www.nature.com/articles/s41586-021-04301-9>.
- Galo Gallardo, Guillermo Rodríguez-Llrente, Lucas Magariños Rodríguez, Rodrigo Morant Navascués, Nikita Khvatkin Petrovsky, Rubén Lorenzo Ortega, and Roberto Gómez-Espinosa Martín. Differentiable Deep Learning Surrogate Models Applied to the Optimization of the IFMIF-DONES Facility. *Particles*, 8(1):21, February 2025. doi: 10.3390/particles8010021.
- Roberto Kersevan and Marton Ady. Recent developments of Monte-Carlo codes Molflow+ and Synrad+, 2019. URL <https://cds.cern.ch/record/2694236>.
- Annika Lang and Jürgen Potthoff. Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3), January 2011. ISSN 1569-3961. doi: 10.1515/mcma.2011.009. URL <http://dx.doi.org/10.1515/MCMA.2011.009>.
- Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv e-prints*, art. arXiv:2010.08895, October 2020. doi: 10.48550/arXiv.2010.08895.
- Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL <http://jmlr.org/papers/v22/20-1364.html>.
- Guillermo Rodríguez-Llrente, Galo Gallardo Romero, and Roberto Gómez-Espinosa Martín. Applications of fourier neural operators in the ifmif-dones accelerator. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024. URL <https://openreview.net/forum?id=FL6ePnpBhB>.
- A. Sabogal, B. Others, and C. Others. Multipurpose vacuum accident scenarios (muvacas) prototype for the ifmif-dones linear accelerator. In *Proc. IPAC'23*, number 14 in IPAC'23 - 14th International Particle Accelerator Conference, pages 4324–4327. JACoW Publishing, Geneva, Switzerland, 05 2023. ISBN 978-3-95450-231-8. doi: 10.18429/JACoW-IPAC2023-THPA156. URL <https://indico.jacow.org/event/41/contributions/2360>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2024. URL <https://arxiv.org/abs/2407.17032>.

Xiaodi Wang, Youbo Liu, Junbo Zhao, Chang Liu, Junyong Liu, and Jinyue Yan. Surrogate model enabled deep reinforcement learning for hybrid energy community operation. *Applied Energy*, 289: 116722, 2021. doi: 10.1016/j.apenergy.2021.116722.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *CoRR*, abs/2009.13303, 2020. URL <https://arxiv.org/abs/2009.13303>.

A Iterative evaluation of the surrogate model

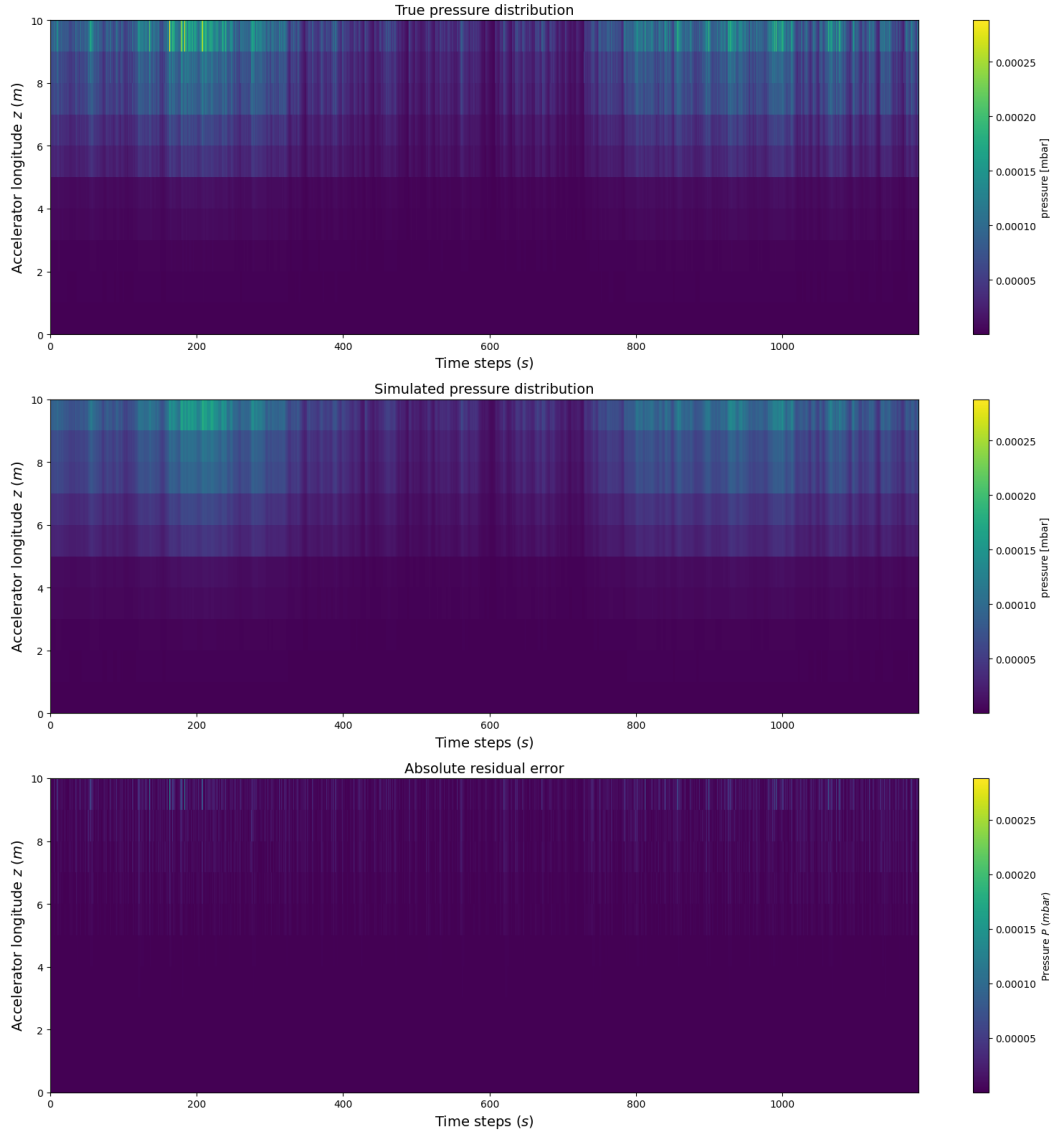


Figure 2: Iterative evaluation through time of the FNO surrogate model. It takes an initial pressure distribution along the accelerator longitude (in arbitrary units, each point represents a sensor) and different argon injections in each step. The vertical axis in each plot represents z , the colorbar represents the pressure p and the horizontal axis is the time t . From top to bottom: the real pressure distributions from the test dataset, the simulated by the model pressure distributions and the residual errors.

B Hyperparameters of the FNO surrogate model

Table 1: NVIDIA Modulus FNO architecture hyperparameters. Software versions: `nvidia-modulus` 0.6.0 and `nvidia-modulus.sym` 1.5.0, Ubuntu 22.04.4 LTS (Docker), and Torch 2.3.0.

Hyperparameter	Value	Description
<code>scheduler</code>	<code>tf_exponential_lr</code>	Learning rate schedule type
<code>optimizer</code>	<code>adam</code>	Optimization algorithm
<code>loss</code>	<code>sum</code>	Loss aggregation method
<code>decoder.nr_layers</code>	1	Number of decoder layers
<code>decoder.layer_size</code>	256	Hidden units per decoder layer
<code>fno.dimension</code>	1	Spatial dimension of the problem
<code>fno.nr_fno_layers</code>	6	Number of FNO layers
<code>fno.fno_modes</code>	12	Retained Fourier modes per layer
<code>scheduler.decay_rate</code>	0.95	Exponential LR decay factor
<code>scheduler.decay_steps</code>	1000	Steps before each LR decay
<code>training.max_steps</code>	10000	Total training iterations
<code>batch_size.grid</code>	32	Batch size for training data
<code>batch_size.validation</code>	32	Batch size for validation data

C Variables and hyperparameters of the DRL problem

Table 2: Observation and action space variables with their type in Gymnasium, ranges, and descriptions.

Variable	Type	Range	Description
p_t	Box(1,)	$[-6, -3]$	Current pressure ($\log_{10}(\text{mbar})$) at sensor 12
p_{obj}	Box(1,)	$[-6, -3]$	Target pressure ($\log_{10}(\text{mbar})$)
q_t	Box(1,)	$[0.0, 1.0]$	Argon injection rate (sccm, normalized)
action	Box(1,)	$[0.01, 0.9]$	Action value: new argon injection rate

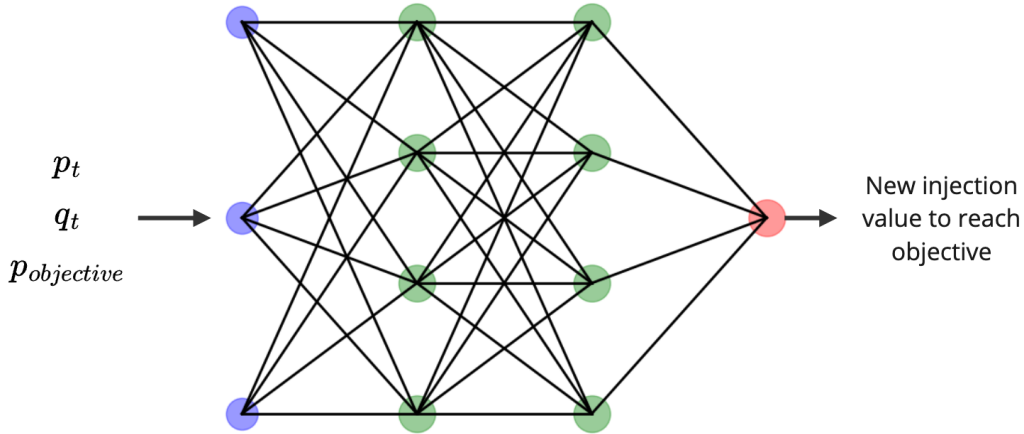


Figure 3: Policy neural network. Its inputs and output are described in Table 2.

Table 3: PPO hyperparameters used with Stable-Baselines3. The agent training was stopped at 270k steps.

Hyperparameter	Value	Description
policy	MultiInputPolicy	Policy network type (supports dict/obs inputs)
learning_rate	0.0008	Step size for gradient descent
n_steps	2048	Rollout steps per update
batch_size	64	Minibatch size for each gradient update
n_epochs	10	Number of epochs per update
gamma	0.902	Discount factor for rewards
gae_lambda	0.95	GAE parameter for bias–variance trade-off
clip_range	0.2	PPO clipping parameter
clip_range-vf	null	Clipping for value function (disabled)
normalize_advantage	true	Advantage normalization enabled
ent_coef	0	Entropy bonus coefficient
vf_coef	0.5	Value function loss coefficient
max_grad_norm	0.5	Gradient clipping norm
total_timesteps	1,000,000	Training budget in timesteps
act_function	tanh	Activation function in policy network
layers	[64, 64]	Policy and value network architecture

D Agent evaluation within the digital environment

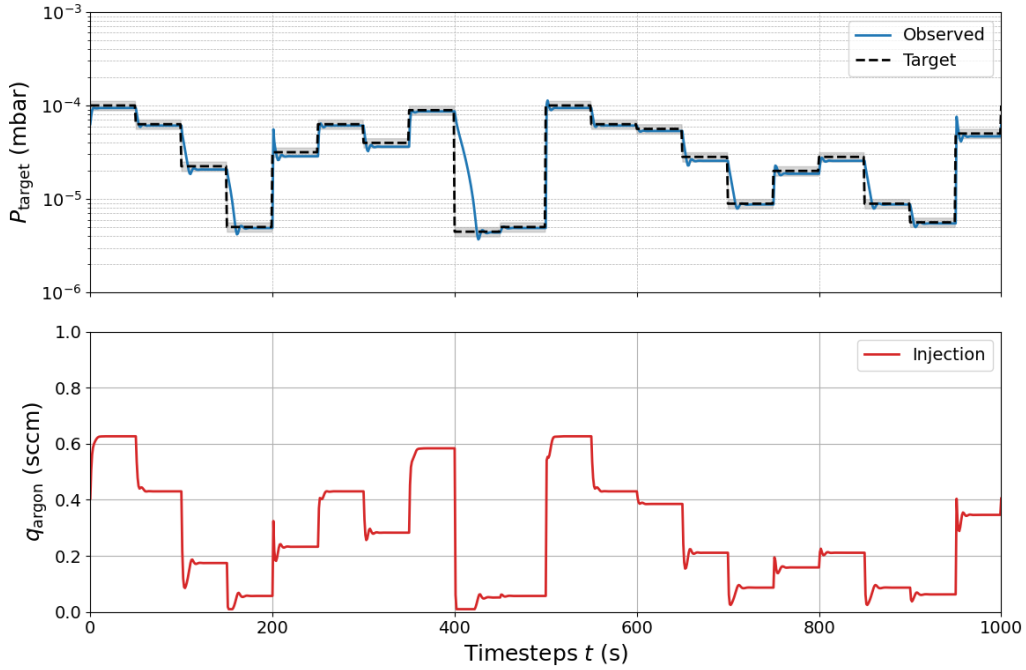


Figure 4: Evaluation of the DRL agent on a different simulated environment. Top: observed (achieved by the agent) and objective (reference) pressures. Bottom: argon injection values.

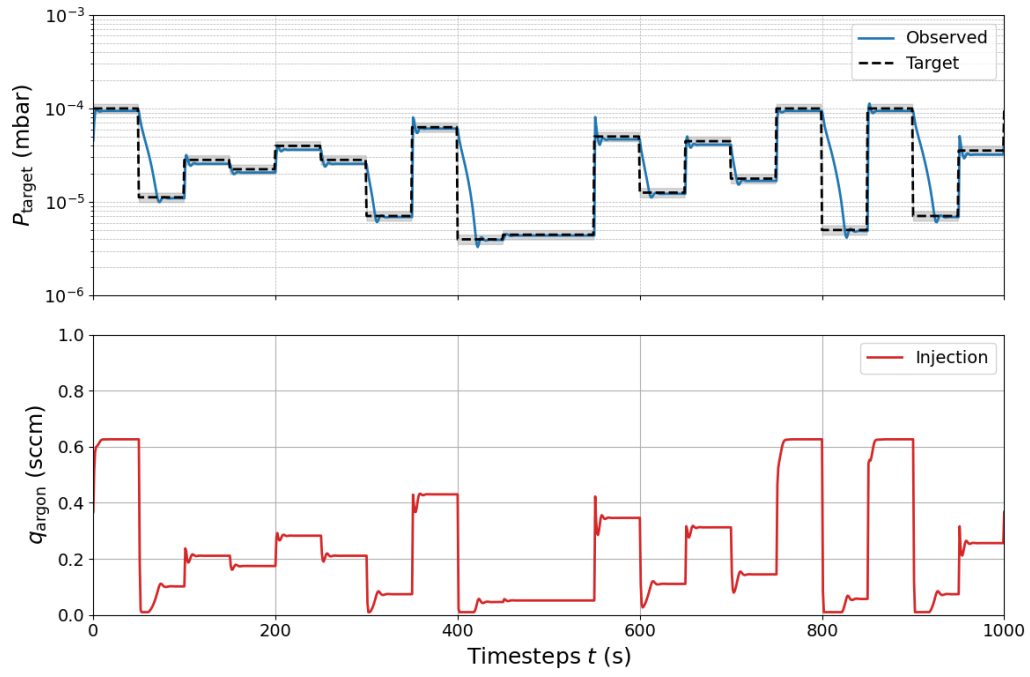


Figure 5: Evaluation of the DRL agent on a different simulated environment. Top: observed (achieved by the agent) and objective (reference) pressures. Bottom: argon injection values.

E Agent evaluation within the real prototype

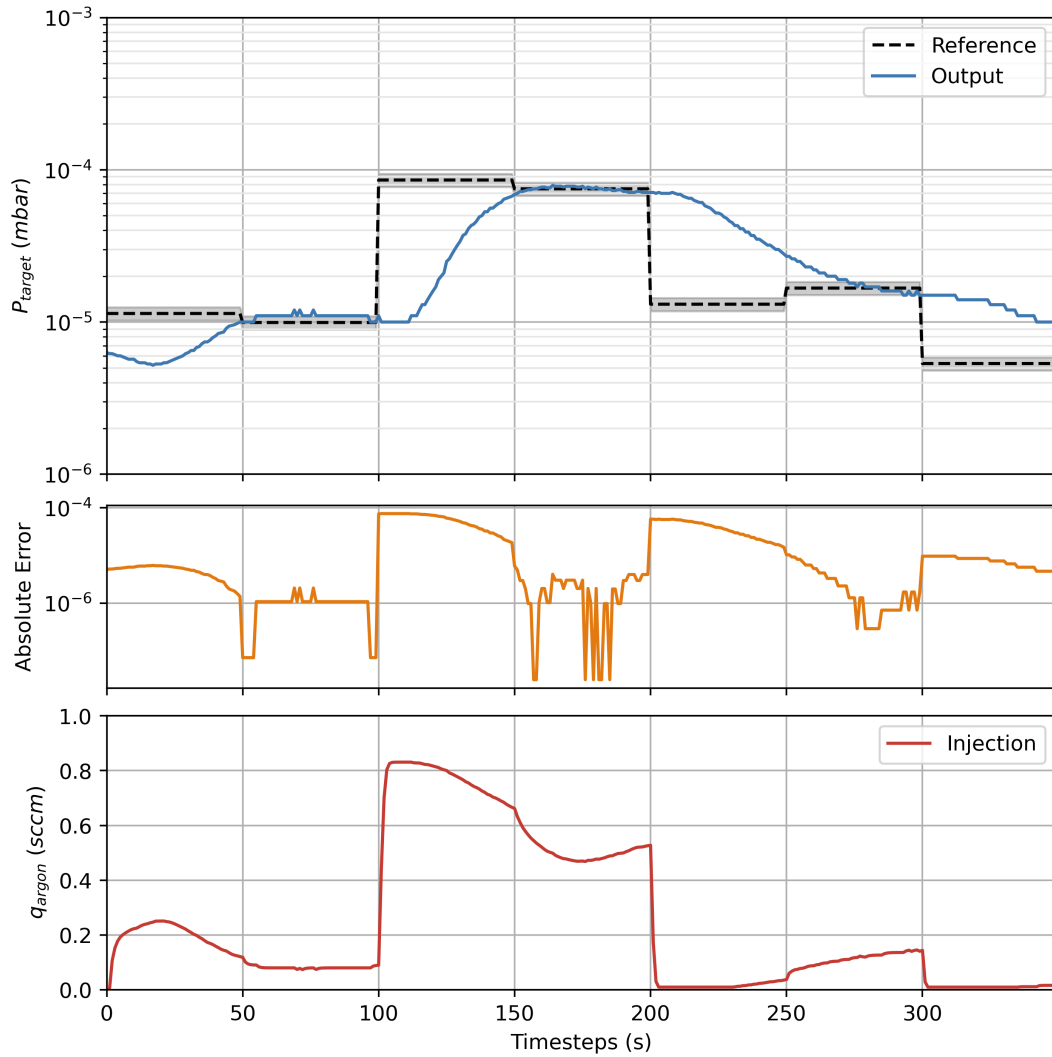


Figure 6: Evaluation of the DRL agent on the real prototype. Top: measured and objective pressures. Middle: absolute error between objective and achieved pressures. Bottom: argon injection rates. In this test, the transition interval between target pressure changes is set to 50 seconds. Because of the injector's position in the system, the agent cannot fully reach the target within this time frame. Extending the transition interval, however, enables the agent to achieve the desired pressure.

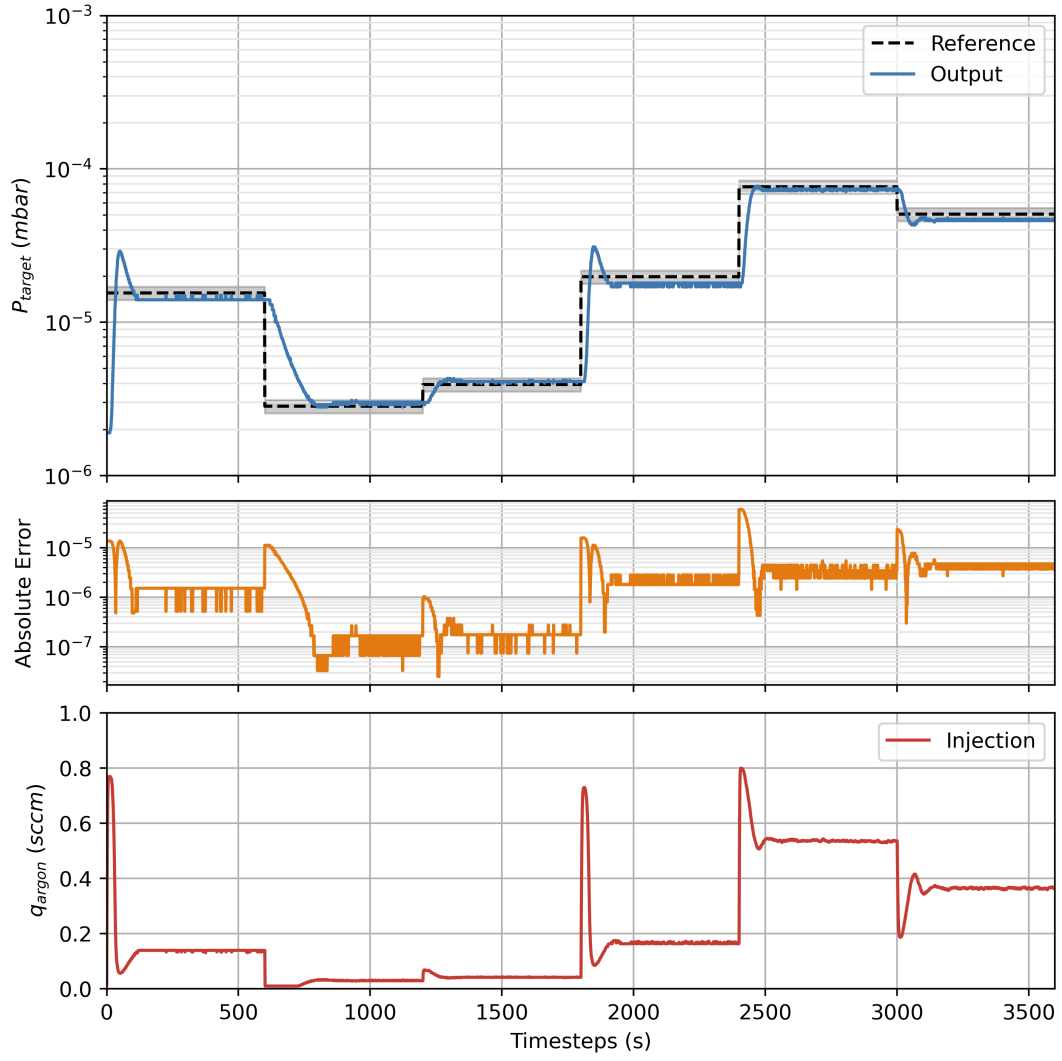


Figure 7: Evaluation of the DRL agent on the real prototype. Top: observed and objective pressures. Bottom: argon injection values.