Fourier-Neural-Operator-based hybrid reinforcement learning framework for continuous adaptive control*

Oleksandr Petrovskyi[1,*,†] and Serge Dolgikh[2,†]

[1] *Exoft LLC, T. Shevchenka 111A, Lviv, Ukraine*
[2] *Solana Networks, 15 Fitzgerald Road, Ottawa, Canada*

Abstract

Fourier Neural Operator has recently emerged as an efficient tool for working with complex differential-equation-based systems. Its unique features, such as differentiability, resolution invariance, and forward-pass efficiency, make it a highly appealing choice as a world model for hybrid offline-to-online reinforcement learning methods in the context of adaptive control of complex continuous industrial processes. However, to our knowledge, this has not been currently researched, so this study aims to address this gap by designing a multi-stage offline-to-online hybrid reinforcement learning framework that utilizes the advantages of Fourier neural operators to increase the stability and speed of convergence of an agent, while avoiding common drawbacks of pure model-based methods. We formulate the framework, justify its advantages, and describe the procedure for numerical experiments to demonstrate its validity and effectiveness compared with other popular methods. Results of the numerical experiment will be presented in future work.

Keywords

adaptive control, reinforcement learning, Fourier Neural Operator, offline-to-online learning, hybrid reinforcement learning, sim-to-real gap, pretraining, IndPenSim

1. Introduction

Reinforcement learning (RL) is a powerful framework for implementing adaptive control systems that has demonstrated success across many different domains. However, issues such as sample inefficiency, poor generalization, and simulation-reality gaps prevent it from being widely adopted in real-world industrial settings, where exploration is costly and risks are high. To address these challenges, offline-to-online methods and hybrid architectures are being actively researched, as they enable an efficient combination of prior knowledge on the environment with the flexibility of pure RL. Neural operators emerged recently as a powerful scientific machine learning tool specifically designed to learn the underlying partial differential equations from observations in infinite-dimensional functional spaces. Their unique properties, such as the ability to capture complex dynamics, differentiability, resolution invariance, output discretization flexibility allowing zero-shot super-resolution, forward-pass efficiency, and mathematical well-posedness, make them a highly appealing choice as world models in offline-to-online hybrid RL methods in the context of continuous adaptive control tasks. But, to our surprise, this has not yet been studied; thus, this paper aims to address this knowledge gap.

* *SmartIndustry 2026: 3rd International Conference on Smart Automation & Robotics for Future Industry, March 26-27, 2026, Lviv, Ukraine*
* Corresponding author.
† These authors contributed equally.
oleksandr.s.petrovskyi@lpnu.ua (O. Petrovskyi); sdolgikh@lpnu.ua (S. Dolgikh)
0000-0002-5729-544X (O. Petrovskyi); 0000-0001-5929-8954 (S. Dolgikh)

2. Background and Problem Statement

Industrial autonomous control systems are still primarily implemented using proportional integral derivative (PID) or model predictive control (MPC) controllers [1], with the former often yielding suboptimal control policies and being unapplicable to environments with complex dynamics, while the latter requires extensive domain knowledge and thorough manual design. RL has shown great promise in simplifying the process of building highly effective controllers by learning optimal control policies directly from data without the need for explicit modeling. However, their limitations, such as *poor generalization* (shortcuts are learned instead of general principles) [2], *instability* and *sensitivity to hyperparameters* (even a change in random seed may affect the algorithm's convergence) [3, 4], unavoidable *simulation-reality gaps* [5, 6], *low sample efficiency* [4, 5], *safety* and *explainability* [5, 7], *sparse rewards* [4, 8], and the fundamental problem of *exploration-exploitation tradeoff* (we don't want the agent to experiment when its improper decision can cause financial losses or real harm, but at the same time, preventing exploration will guarantee suboptimal control) has rendered pure data-driven approaches hardly applicable [5, 9]. Thus, hybrid methods, which imbue the agent with prior knowledge before it is introduced to the real environment, thereby reducing exploration and speeding up and stabilizing the algorithm's convergence, are actively being researched [10].

Neural operators [11] have recently emerged as an extension of neural networks to infinite-dimensional functional spaces and have been proven useful for a wide range of scientific machine learning tasks. In particular, the Fourier Neural Operator (FNO) [12] networks can be used to obtain highly efficient (from 100 to 1,000,000 times faster than numerical solvers), resolution-invariant, and differentiable surrogate environment models by learning direct mappings between input condition functions and solution functions of PDE purely from historical data [13], and were successfully applied in the real world for forecasting weather [14], industrial robot load detection in a manufacturing environment [15] and many more. The use of FNO in adaptive control is being actively researched for its ability to capture the dynamics of complex systems. In particular, the DeepONet architecture was extended to effectively predict the medium/long-term dynamic response of non-autonomous systems with time-dependent inputs [16]; FNO-based controllers were proposed in [17]; a promising Fourier Q Operator Network (FQON) architecture was implemented and validated in an industrial setting [18]; an advanced FNO-based predictive maintenance solution was developed in [19].

However, FNO also has the potential to be used in the context of offline-to-online RL, as their ability to capture complex dynamics, focus on differential-equation-driven systems, continuous nature, differentiability, resolution invariance, and computational efficiency make them highly appealing to use as world models; thus, our study aims to cover this gap.

3. Proposed Method

3.1. Method Description

Given that historical or synthetic training data is available in sufficient amounts (which is required for any form of pretraining to be possible), the proposed method lies in first training an FNO-based-network W to predict the next state s_{t+1} based on $h \in \mathbb{N}$ previous observations $s_{t-h \ldots t}$ and actions taken $a_{t-h \ldots t}$. After that, the network's weights are frozen, W is integrated into an RL agent as a world model, and, after a warm-up training where W is reused as an environment substitute, the agent can be deployed to the real system, requiring fewer exploration steps for achieving stable and effective control.

To restrain model-free trial-and-error learning, which is undesirable in high-risk industrial environments, and mitigate simulation-reality mismatches or possible exploitation of the FNO approximation, environment dynamics knowledge is introduced into the model-free agent as a model-based branch with the actor loss having the form of (1), with L_{TD3} being a standard Critic-based TD3 loss, L_{FNO} being an analytical loss, that utilizes Dreamer-like imaginary environment rollouts and differentiability of FNO, and $\lambda \in (0,1)$ being a hyperparameter balancing the two.

$$L_{Actor} = \lambda \cdot L_{TD3} + (1 - \lambda) \cdot L_{FNO} \quad (1)$$

Despite introducing computational complexity, this approach enables a hybrid architecture that mitigates the initial poor performance and sample inefficiency of model-free methods, as well as simulation-reality mismatch and value overestimation of model-based techniques. Twin Delayed Deep Deterministic Policy Gradient (TD3) [20] was chosen as the base algorithm for this study due to its deterministic policy with a controlled noise term, as a stochastic policy would negatively impact multi-branch gradient computation.

Offline Stage. Let D be a dataset of historical observations in the form of an array of 2-tuples (s_t, a_t) defined as (2), where $s_t \in S$ is a state observed at timestep t and $a_t \in A$ is an action taken at that timestep.

$$D = [(s_t|a_t)]_{t=1}^{T} = [(s_1, a_1), (s_2, a_2), \dots (s_T, a_T)] \quad (2)$$

By splitting D into pairs of $([(s_{t-h}, a_{t-h}), (s_{t-h+1}, a_{t-h+1}) \dots (s_t, a_t)], s_{t+1})$, where h is the size of the history, an FNO-based network W can be trained to predict the next state s_{t+1} based on prior observations s_t and actions taken a_t, as illustrated in (3).

$$W([(s_i, a_i)]_{i=t-h}^{t}) \mapsto s_{t+1} \quad (3)$$

The ability to use neural operators for middle/long-term controlled system response prediction was described and validated in [16].

Warm-up Stage. Upon receiving a well-trained model W, its weights are frozen and integrated into an RL agent, as shown in Picture 1. Then, warm-up training episodes are run with a detached instance of W used as a substitute for the real environment. This step allows us to obtain an initial policy using already available data without requiring additional simulations. However, heavy regularization measures (e.g., uncertainty quantification) must be implemented at this stage to prevent the network from overfitting to the surrogate environment, which is a topic for future study.

Inference. After this, the agent can be integrated into the real environment, converging faster due to being pretrained on the environment approximation and able to perform imaginary rollouts, lowering the risk of bringing the system to highly undesirable states. The world model W is periodically updated in accordance with new observations, separately from the agent, using the same procedure as in the Offline Stage.

3.2. Method Justification

Several features of FNO make the proposed approach promising.

Continuous nature. Most real-world processes (especially in biotechnology, robotics, etc.) are governed by complex differential equation systems. In many cases, approximating their dynamics with harmonic functions may be more natural than folding, stretching, and trimming the input space, as regular neural networks do. As a result, FNO was shown to better capture the structural dynamics than LSTM in both linear and non-linear settings [21]. Also, as they operate on functions rather than data points, FNOs can process data at various resolutions. This allows the use of a wider range of historical data, which determines the feasibility of controller implementation. Neural operators, trained on low-resolution data, can be applied to high-resolution data, which is often the case in real-world applications. There are also extensions of regular neural operators that enable processing data with irregular sampling rates [22], which further increases the applicability of the proposed approach.

Differentiability. Operating in functional space yields continuous functions as FNO's output instead of traditional $\mathbb{R}^n \to \mathbb{R}^n$ data point mapping, which allows direct backpropagation through the entire family of approximate environment models rather than a single neural network approximation.

Performance of FNO. Once trained, surrogate models obtained via FNO were shown to be 100–1,000,000 times faster than ordinary PDE solvers [13]. This makes FNO appealing for use as a world model in model-based RL algorithms, allowing many Dreamer-style imaginary environment rollouts to be easily performed, which, in combination with the abovementioned differentiability, results in *backpropagation through time* and, thus, leads to more far-sighted planning, potentially tackling the sparse-reward issue.

4. Experiment Design

4.1. Environment

The proposed approach will be evaluated on PenSimPy [23] environment – a Python implementation of IndPenSim [24], a highly realistic mathematical model of a fed-batch *Penicillium chrysogenum* fermentation that is widely used as a benchmark for advanced control and monitoring methods. This environment was developed based on a mechanistic model that has been shown to best describe the fermentation behavior when compared with ten other penicillin models and validated using historical data collected from a real industrial-scale penicillin fermentation process. Its high complexity, non-linearity and partial observability make it a suitable testbed for evaluating the performance of the proposed approach. The task performed by the agent is the maximization of batch yield (the amount of useful product produced during an episode), with a maximum known value being around 4000 kg for a 100,000-liter vessel. Sampling of observable variables and decision making is performed every 12 minutes. The duration of each episode is 1151 steps, which is equivalent to 230 hours in real life.

4.2. Compared Algorithms

To validate the effectiveness of our approach, it will be compared against its non-operator version and 3 other popular methods: pure RL (TD3), offline RL (CQL+TD3), and a reference MPC-controller.

Advanced MPC. Results obtained using an advanced MPC controller, designed by the authors of IndPenSim [25], will be used as a gold standard as the controller managed to reach a stable 30.13 g/l biomass concentration while reducing the process variability along multiple runs.

Pure RL. Comparing the performance of our approach against a regular TD3 will serve as an ablation study, allowing us to assess the degree of reduction in exploration needs and increase in convergence speed and stability, achieved by embedding the prior knowledge of environmental dynamics into an agent.

Non-Operator Version. To gain a deeper understanding of the benefits provided by the neural-operator-based world model, a non-operator version of the method (with FNO replaced by an ordinary deep neural network) will be studied, to ensure that FNO is beneficial for capturing the complex dynamics of the environment.

Offline RL. The approach will also be compared against a popular offline RL technique - Conservative Q-Learning [26], which allows for achieving a good initial control policy from historical data, that can be further integrated into TD3, while avoiding the issue of value overestimation. This is necessary to assess the limitations of traditional algorithms in complex nonlinear environments and the potential computational overhead introduced by a world model with imaginary rollouts.

For every method, the error between the current y_t and the highest possible y^* biomass concentration will be used to evaluate the course of the control process, with *Mean Absolute Error* MAE $\stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T}|y^* - y_t|$ representing the cumulative divergence between the real and the desired state, *Mean Squared Error* MSE $\stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T}(y^* - y_t)^2$ capturing the strength of deviations from the target concentration, *Integral Time Absolute Error* ITAE $\stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T} t \cdot |y^* - y_t|$ measuring the ability of the controller to promptly bring the system to the desired state and *Integral Time Squared Error* ITSE $\stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T} t \cdot (y^* - y_t)^2$ showing the ability of the controller to maintain a system in a stable state. Further details on technical implementation, hyperparameter tuning, and results of the numerical experiments will be presented in future work.

5. Conclusion

This study identified a gap in the state of the art in adaptive control, particularly for complex continuous control tasks, and proposed a novel approach to address it using FNO networks. We formulated and justified the conceptual framework for deriving a surrogate world model from available historical data via an FNO-based-network, integrating it into the TD3 algorithm as an additional model-based branch to allow the agent to leverage prior knowledge while avoiding common model-based RL limitations, warming up the agent using the surrogate model as an environment substitute, and integrating it into the real system. We defined the concept and procedure of the numerical experiment comparing the proposed approach with conventional methods, as well as designed and implemented the experimental testbed. The results of the experiment will yield important inputs about the potential benefits of using pretrained neural operator networks as parts of world models in hybrid reinforcement learning algorithms. This research lays the groundwork for future studies in this area, with potential applications across various industries where continuous adaptive control is essential.

Declaration on Generative AI

During the preparation of this work, authors used Grammarly for grammar and spellchecking.

6. References

[1] K. Agyei, P. Sarhadi, and D. Polani, “Deep Reinforcement Learning in Applied Control: Challenges, Analysis, and Insights,” Jul. 10, 2025, *arXiv*: arXiv:2507.08196. doi: 10.48550/arXiv.2507.08196.

[2] Q. Delfosse *et al.*, “Deep Reinforcement Learning Agents are not even close to Human Intelligence,” May 27, 2025, *arXiv*: arXiv:2505.21731. doi: 10.48550/arXiv.2505.21731.

[3] J. Bjorck, C. P. Gomes, and K. Q. Weinberger, “Is High Variance Unavoidable in RL? A Case Study in Continuous Control,” Feb. 05, 2022, *arXiv*: arXiv:2110.11222. doi: 10.48550/arXiv.2110.11222.

[4] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, “Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, no. Volume 8, 2025, pp. 153–188, May 2025, doi: 10.1146/annurev-control-030323-022510.

[5] A. Farooq and K. Iqbal, “A Survey of Reinforcement Learning for Optimization in Automation,” in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, Aug. 2024, pp. 2487–2494. doi: 10.1109/CASE59546.2024.10711718.

[6] T. Zhang and H. Mo, “Reinforcement learning for robot research: A comprehensive review and open issues,” *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 17298814211007305, May 2021, doi: 10.1177/17298814211007305.

[7] Y. M. Alginahi, O. Sabri, and W. Said, "Reinforcement Learning for Industrial Automation: A Comprehensive Review of Adaptive Control and Decision-Making in Smart Factories," *Machines*, vol. 13, no. 12, Dec. 2025, doi: 10.3390/machines13121140.

[8] S. Chakraborty, A. S. Bedi, A. Koppel, P. Tokekar, and D. Manocha, "Dealing with Sparse Rewards in Continuous Control Robotics via Heavy-Tailed Policies," Jun. 12, 2022, *arXiv*: arXiv:2206.05652. doi: 10.48550/arXiv.2206.05652.

[9] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of Real-World Reinforcement Learning," Apr. 29, 2019, *arXiv*: arXiv:1904.12901. doi: 10.48550/arXiv.1904.12901.

[10] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 10237–10257, Aug. 2024, doi: 10.1109/TNNLS.2023.3250269.

[11] N. Kovachki *et al.*, "Neural Operator: Learning Maps Between Function Spaces," May 02, 2024. doi: 10.5555/3648699.3648788.

[12] Z. Li *et al.*, "Fourier Neural Operator for Parametric Partial Differential Equations," May 17, 2021, *arXiv*: arXiv:2010.08895. doi: 10.48550/arXiv.2010.08895.

[13] V. Duruisseaux, J. Kossaifi, and A. Anandkumar, "Fourier Neural Operators Explained: A Practical Perspective," Jan. 22, 2026, *arXiv*: arXiv:2512.01421. doi: 10.48550/arXiv.2512.01421.

[14] B. Bonev *et al.*, "FourCastNet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale," Jul. 18, 2025, *arXiv*: arXiv:2507.12144. doi: 10.48550/arXiv.2507.12144.

[15] F. Gao, D. Xie, and Y. Xie, "Load detection of industrial robots in manufacturing environment based on improved FNO network," *Front. Mech. Eng.*, vol. 11, Nov. 2025, doi: 10.3389/fmech.2025.1666911.

[16] G. Lin, C. Moya, and Z. Zhang, "On Learning the Dynamical Response of Nonlinear Control Systems with Deep Operator Networks," Sep. 26, 2023, *arXiv*: arXiv:2206.06536. doi: 10.48550/arXiv.2206.06536.

[17] L. Bhan, Y. Shi, and M. Krstic, "Operator Learning for Nonlinear Adaptive Control," in *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, PMLR, Jun. 2023, pp. 346–357. Accessed: Feb. 03, 2026. [Online]. Available: https://proceedings.mlr.press/v211/bhan23a.html

[18] Y. Ni, Y. Li, C. Liu, and Y. Jin, "A New Fourier Q Operator Network Based Reinforcement Learning Method for Continuous Action Space Decision-making in Manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102641, Apr. 2024, doi: 10.1016/j.rcim.2023.102641.

[19] S. Qiu, "Optimizing Predictive Maintenance in Intelligent Manufacturing: An Integrated FNO-DAE-GNN-PPO MDP Framework," Nov. 05, 2025, *arXiv*: arXiv:2511.05594. doi: 10.48550/arXiv.2511.05594.

[20] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," arXiv.org. Accessed: Feb. 10, 2026. [Online]. Available: https://arxiv.org/abs/1802.09477v3

[21] R. Haghi, B. Gaikwad, and A. Patra, "Fourier Neural Operators for Structural Dynamics Models: Challenges, Limitations and Advantages of Using a Spectrogram Loss," Nov. 11, 2025, *arXiv*: arXiv:2511.08753. doi: 10.48550/arXiv.2511.08753.

[22] Y. Li and S. Zhe, "Graph-Based Operator Learning from Limited Data on Irregular Domains," May 25, 2025, *arXiv*: arXiv:2505.18923. doi: 10.48550/arXiv.2505.18923.

[23] M. Zhang *et al.*, "SMPL: Simulated Industrial Manufacturing and Process Control Learning Environments," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26631–26644, Dec. 2022.

[24] S. Goldrick, A. Ştefan, D. Lovett, G. Montague, and B. Lennox, "The development of an

industrial-scale fed-batch fermentation simulation," *Journal of Biotechnology*, vol. 193, pp. 70–82, Jan. 2015, doi: 10.1016/j.jbiotec.2014.10.029.

[25] C. A. Duran-Villalobos, S. Goldrick, and B. Lennox, "Multivariate statistical process control of an industrial-scale fed-batch simulator," *Computers & Chemical Engineering*, vol. 132, p. 106620, Jan. 2020, doi: 10.1016/j.compchemeng.2019.106620.

[26] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," Aug. 19, 2020, *arXiv*: arXiv:2006.04779. doi: 10.48550/arXiv.2006.04779.