Staged integration of Recurrent Soft Actor-Critic for bioreactor control[*]

Oleksandr Petrovskyi[1,*,†] and Yevgen Martyn[2,†]

[1] *Lviv Polytechnic National University, S. Bandery 12, 79013, Lviv, Ukraine*
[2] *Lviv State University of Life Safety, Kleparivska 35, 79007, Lviv, Ukraine*

Abstract

Bioreactors play a crucial role in making industrial biotechnological processes possible by providing a tool for creating and maintaining an optimal environment for proteins, cells, and cell cultures to grow and function in. In most cases, the overall technical and economic performance of an industrial biotechnological process heavily depends on the efficiency of utilized bioreactors; thus, optimizing their operation is of great theoretical and practical interest. However, due to the highly complex and stochastic nature of bioprocesses, many issues arise during the development and implementation of their autonomous control systems.

This article analyzes the main challenges associated with the development of autonomous bioreactor controllers, reviews the most prominent ways of tackling them with reinforcement learning, and presents an implementation of an offline-to-online memory-based RL controller applied for the custom simulation of a backer yeast fed-batch bioreactor with a partially observable state.

Results show that pretraining on approximate simulations can be successfully applied to increase the generalization capabilities and convergence speed of a memory-based RL agent in the context of partially observable bioprocess control, reducing the time required to reach high rewards. However, such questions as avoiding harmful overfitting during the pretraining and implementing an efficient memory mechanism for an agent remain open.

Keywords

bioprocess control, fed-batch bioreactor, reinforcement learning, offline-to-online, RSAC, LSTM, POMDP

1. Introduction

Bioreactors are indispensable in industrial biotechnology: they are used to produce vaccines, antibiotics, biofuels, food, and beverages, synthesize complex proteins and enzymes, process waste, grow tissues, organs, and more [1, 2, 3]. In the vast majority of cases, it is the efficiency of the bioreactor that has a decisive impact on the overall technical and economic performance of the production process [4]. That is why creating flexible and effective bioreactor control systems is a pressing problem the solution of which can affect many areas of human life.

However, in practice, a functioning bioreactor is an extremely complex stochastic dynamic system with a high degree of nonlinearity. Many factors can influence the course of a bioprocess both at the macroscopic level (substrate absorption, oxygen saturation, accumulation of growth inhibitors, etc.) and at the microscopic level of individual cells [5, 6]; there are significant limitations in the ability to measure the actual state of the system [6]; each bioprocess requires its specific parameters to be taken into account, and each bioreactor is different from the others, as it is specially designed to solve particular tasks [3]. In addition, we never know the exact model of the underlying bioprocess, and finding

[*] *2nd International Conference on Smart Automation & Robotics for Future Industry, April 03-05, 2025, Lviv, Ukraine*
[1*] Corresponding author.
[†] These authors contributed equally.

oleksandr.s.petrovskyi@lpnu.ua (O. Petrovskyi); evmartyn@gmail.com (Ye. Martyn)
0000-0002-5729-544X (O. Petrovskyi); 0000-0001-9095-7057 (Ye. Martyn)

an approximate model that aligns with the limited experimental data can be quite challenging [7]. That is why such modern process control methods as Proportional-Integral-Derivative (PID), Fuzzy Logic Control (FLC), and Model Predictive Control (MPC), despite their widespread use, are often unable to provide optimal control of bioprocesses due to their limitations or require an accurate mathematical model, which in most cases is extremely difficult to build [8].

This has led to growing scientific interest in reinforcement learning (RL) in the context of bioreactor control [9, 10, 11], thanks to which the optimal control problem can be considered as a Markov Decision Process (MDP) in which an agent tries to learn to maximize the cumulative reward it receives when interacting with a complex, uncertain environment [12]. Extensive experience in applying RL in various industrial environments has shown that RL controllers are able to outperform traditional methods in control accuracy and speed, have an outstanding ability to generalize and learn without prior knowledge and models built by experts [13].

Therefore, this article aims to review the modern experience of using reinforcement learning in the context of autonomous bioreactor control, as well as common related problems and ways to solve them, synthesize a practical approach to implementing an RL-controller, and test it on the simulation of a fed-batch baker yeast bioreactor with a partially observable state.

2. Related Work

Treloar et al. [14] investigated the possibility of utilizing DQN to effectively control bioprocesses where several microbial cultures coexist simultaneously. The authors empirically proved the algorithm's robustness to different initial conditions, demonstrated the possibility of obtaining a sufficiently good policy in 24 hours with the help of the parallel use of 5 bioreactors, and even showed the ability of an RL agent to perform control with higher efficiency than a traditional PID controller as measurement frequency decreases.

Authors of [15] have successfully applied Deep Deterministic Policy Gradient (DDPG) to control the temperature of an ethanol fermenter (simulation), demonstrating faster convergence and higher control precision in comparison with DQN, as well as its ability to effectively react to random disturbances in the force and temperature of the incoming flow, quickly returning the system to the desired state. Several sources describe the successful application of an improved version of DDPG – T3D (Twin Deep Delayed Deterministic Policy Gradient) for controlling bioreactors and wastewater treatment systems (simulations) [16, 17, 18] and its ability to converge to better policies.

Soft Actor-Critic is a widely used RL algorithm that often demonstrates state-of-the-art performance [19]. SAC effectively works with continuous observation and action spaces, utilizes the *stochastic policy*, which increases the robustness to uncertainties, and the *maximum entropy framework* with automatic temperature adjustment, which provides a more efficient environment exploration strategy. In addition, it uses all the successful architectural decisions of the algorithms discussed above: off-policy learning, replay buffer, Actor-Critic design, double Q-functions, target networks, soft update, etc.

Performance comparison of PG, DQN, A2C, DDPG, and SAC algorithms applied to such bioreactor control tasks as valuable product maximization and maintaining the biomass concentration at a fixed level demonstrated that the Actor-Critic architecture is significantly more effective than value-based and policy-based methods separately, and among all the algorithms implementing it, SAC had the best performance in terms of the convergence speed and control efficiency [20]. In another study, SAC outperformed TRPO, PPO, and TD3 in the task of HVAC control [21].

In practice, despite the availability of powerful algorithms, there is often a lack of available data for offline training of RL controllers, while the cost of online training is too high. The situation is further complicated by the complexity of bioprocesses, which makes it challenging to create high-quality mechanistic models and simulations of them. Therefore, hybrid approaches that can provide a compromise by combining the best of mechanistic and data-driven approaches are of particular interest, despite the fact that, as pointed out by Monteiro and Kontoravdi, “yet it is unclear how best to integrate these two components and how to account for plant-model mismatch that characterizes bioprocesses” [9].

One promising way to solve this problem is offline-to-online reinforcement learning, which combines the stage of offline pretraining on available data or approximate simulation with online adaptation in the real system. And while pretraining is widely used in other areas of ML, offline-to-online reinforcement learning is a relatively new area with many unique problems that are being explored in detail in [22].

In the context of this paradigm, an interesting approach is proposed by Pandian et al. [23]. The authors combine inverse neural networks (INN) and RL, but unlike previous works on similar topics, the separately trained inverse network is not used as a direct policy-function but as a tool to initialize the agent's Q-table. This hybrid approach simultaneously reduces the requirements for the amount of existing data (a disadvantage of INN) and speeds up the convergence of the agent. The authors demonstrated the effectiveness of the described approach in a real-world laboratory setup. However, the disadvantages of the proposed algorithm are the requirements to use tabular Q-Learning and discretize continuous variables, which makes it hard to apply for solving more complex continuous control problems due to the so-called curse of dimensionality.

Petsagkourakis et al. [24] also propose the use of the staged integration of the RL controller, in which, in the first stage, a simple mechanistic approximation of the target process model is used for offline pretraining of the Policy-Gradient agent; in the second, some network parameters are frozen and the network is additionally trained on data obtained from the real system; and in the third, the pretrained RL controller is used to control a real system in online mode.

Another critical problem that arises in the development of autonomous bioprocess controllers is the inability to fully measure the actual state of the system. [6, 25]. This transforms the control task from MDP into Partially Observable Markov Decision Process (POMDP), which is formulated as a 6-tuple , where are defined as in MDP, O is a set of possible observations, and is an emission function that determines which observations are available for an agent, given the current state and the chosen action. Thus, the RL agent is faced with finding an optimal policy that maximizes the expected sum of rewards under conditions of incomplete information.

The dominant way to solve this problem is via the memory-based reinforcement learning – augmentation of the RL agent with a sequence model, mainly in the form of a recurrent neural network [26], or a transformer [27], which provides the agent with “memory”, thanks to which it can approximate the system’s dynamics and the current state based on the history of observations. Some of the algorithms developed with this principle in mind are LSTM-TD3 [28], BSAC [29], and an open-sourced family of recurrent versions of the most popular RL algorithms, which are described and compared in [30]. Out of the latter, RSAC with LSTM layers demonstrated the best performance.

Therefore, considering the effectiveness of the offline-to-online training approach and the ability of memory-based RL algorithms to capture the latent system's dynamics, as well as the advantages of Soft Actor-Critic (and the frequent emergence of new modifications of this algorithm), the problem of developing a method for the staged integration of RSAC for autonomous bioreactor control becomes relevant.

3. Combining Offline-to-online and POMDP

3.1. Staged Integration

The staged integration of the Recurrent Soft Actor-Critic bioreactor controller is proposed to be conducted in the following way:

1. Convert the approximate mechanistic model of the target bioprocess to the form of POMDP and construct an RL environment based on it. Many industrial bioreactors are already operated by MPC-controllers that rely on empirically-derived mathematical models, but if the approximate model is missing, it must be created.
2. Create the RSAC agent and tune its hyperparameters, e.g., the number and size of hidden layers, learning rates, trajectory size, rate of weight updates, etc.
3. Pretrain the agent on several deterministic simulations with simpler dynamics or slightly different parameters to promote a better generalization of biomass concentration change laws to obtain a more flexible initial policy.
4. Pretrain the agent on the original simulation with added random noises and disturbances.
5. If real data is available, pretrain the agent using it via loading it into the agent's replay buffer in the form of 4-tuple $(o_t, a_t, o_{t+1}, r_{t+1})$
6. Integrate the RL controller into the real system and for online finetuning.

This approach allows for taking maximum advantage of the available knowledge about the real system. Thanks to the pretraining a good initial approximation of an ideal control policy is obtained, which is able to speed up the algorithm's convergence, improve its robustness to random noises and disturbances, lower the need for costly exploration of the environment and probability of bringing the bioprocess to irreversible critical states, because the agent, thanks to prior knowledge, will avoid completely unpromising actions even during exploration.

3.2. Custom POMDP Environment

Approbation of the proposed method's effectiveness in maintaining a biomass concentration at a fixed level was conducted with the utilization of a fed-batch baker's yeast bioreactor simulation based on a mathematical model described and used by Pandian and Noel in [27].

This model was chosen because despite the small number of its parameters and controlled variables, it still preserves all the characteristic features of a complex nonlinear system, is easy to modify, and is based on the Monod equation, which describes the cellular growth dynamics and is widely used in environment engineering.

Model's dynamics is defined by the system of two coupled differential equations (1), the first of which describes the rate of change of biomass concentration x_1, and the second –

$x_1 = 7.0$ substrate concentration x_2. Table 1 contains the description and values of the model parameters we used during the method's approbation.

$$\begin{aligned}\frac{dx_1}{dt} &= (r - u_1 - \theta_4) \cdot x_1 \\ \frac{dx_2}{dt} &= -\frac{rx_1}{\theta_3} + u_1 \cdot (u_2 - x_2) \\ r &= \frac{\theta_1 x_2}{\theta_2 + x_2}\end{aligned} \quad (1)$$

Considering the problem as a Markov Decision Process, the system's state consists of biomass and substrate concentrations in the vessel $s = (x_1, x_2)$ while the action is substrate concentration in the feed $a = u_2 \in [5,35]$. As in the original work, the reward function R is defined as an absolute error between the real and desired values of biomass concentration (2).

$$R: s_t \mapsto r_t = -|x_1^* - x_1| \quad (2)$$

However, unlike the original implementation of the environment, we remove the assumption that the current substrate concentration in the bioreactor is an observed variable, turning the control task from MDP to POMDP. This way, the set of possible observations becomes

and emission function$O = x_1 \in \mathbb{R}$can be defined as in (3).

$$Z: s_t \mapsto o_t = (x_1, x_2) \mapsto x_1 + \epsilon \sim \mathcal{N}(0, \sigma_Z) \quad (3)$$

where ϵ is random Gaussian noise with mean 0 and standard deviation σ_Z.

Table 1
Parameters of the Mathematical Model of the Bioreactor

Symbol	Meaning	Value
x_1	Biomass concentration (g/l)	Target variable, initial
x_2	Substrate concentration (g/l)	Latent variable, $x_1 = 7.0$ initial
u_1	Dilution factor (h^{-1})	0.1 $x_2 = 0.1$
u_2	Substrate in the feed (g/l)	Action variable,
$\theta_{1..4}$	Other process parameters	$u_2 \in [5,35]$
	Target biomass concentration (g/l)	$0.31, 0.18, 0.55, 0.05$ 7.5
x_1^*		
σ_Z	Observation noise deviation (g/l)	$\sigma_Z \in \mathbb{R}^+$

Also, in contrast with the original work, during training the environment is initialized with random values $x_1 \sim \mathcal{U}(0,35), x_2 \sim \mathcal{U}(0,35)$ at the beginning of every episode, where $\mathcal{U}(a,b)$ is a continuous uniform distribution bounded by the interval $[a,b]$. Such an approach allows the agent to learn how to quickly restore the system's desired state in case of any random disturbances.

4. Results

4.1. Bioprocess simulations

We used three simulations to test our method: one stochastic (TRUE), which imitates the real bioprocess, and two auxiliary deterministic (AUX1, AUX2), parameters of which were slightly changed with respect to TRUE. Simulations were implemented with the help of Gymnasium and odeint function from scipy Python package as a differential equation system solver. Values of the parameters of each simulation are listed in Table 2.

Table 2
Parameters of True and Auxiliary Simulations

Parameter	TRUE	AUX1	AUX2
θ_1	0.31	0.34	0.28
θ_2	0.18	0.16	0.20
θ_3	0.55	0.60	0.50
θ_4	0.05	0.03	0.07
σ_Z	0.05	0	0

4.2. Comparison of MDP- and POMDP-oriented algorithms

At first, to assess the general ability to maintain the biomass concentration at a fixed level under conditions of limited observability, the performance of SAC and RSAC algorithms was compared when applied to fully and partially observable versions of TRUE simulation respectively. In both cases, training lasted 50 episodes with an environment rollout length of 160 and batch/trajectory size of 16. Both environments were randomly initialized at the beginning of each episode to improve the flexibility of the policy and its robustness to disturbances and untypical states of the system.

To evaluate how quick deviation from the desired state can be eliminated in both cases after sufficient training, 1000 test environment rollouts were performed using the frozen latest policies. The mean average error (MAE) between actual and desired biomass concentration was recorded during each rollout. The mean and standard deviation of MAE of 1000 tests are listed in Table 3.

Table 3
MAE of Ideal and Real Biomass Concentration for Agents with Different Observability

Observability	Partial	Full
Mean	1.12404	0.93135
Std. dev.	0.47539	0.50121

4.3. Integration of the pretrained agent

To check the convergence speed of the algorithm with the proposed approach, sequential pretraining of the RSAC agent was performed with the update period of 4 steps (hours) on deterministic environments AUX1 and AUX2 during seven episodes (week equivalent) 24 steps each (day equivalent), and its application to TRUE environment lasting 48 steps.

The 4-hour weight update period was chosen as it was the smallest trajectory size the algorithm could converge with. The higher update frequency decreases the agent's reaction time, which determines how quickly it can adapt to changes in the environment. However, if the update interval is too small, trajectories in the replay buffer become too short for the sequence model to accurately approximate the environment dynamics from.

For comparison, RSAC without pretraining was also applied to the TRUE environment. Figure 2 illustrates the reward dynamics of each algorithm during the online training (left) and the test 200-step rollout of their frozen latest policies (right).

It can be seen that the agent pretrained on AUX1 and AUX2 simulations reached the reward plateau after 6 hours of operation of the simulated bioreactor, which took 14 hours for an agent without pretraining. The performance indicators of the algorithms with and without pretraining are given in Table 4.

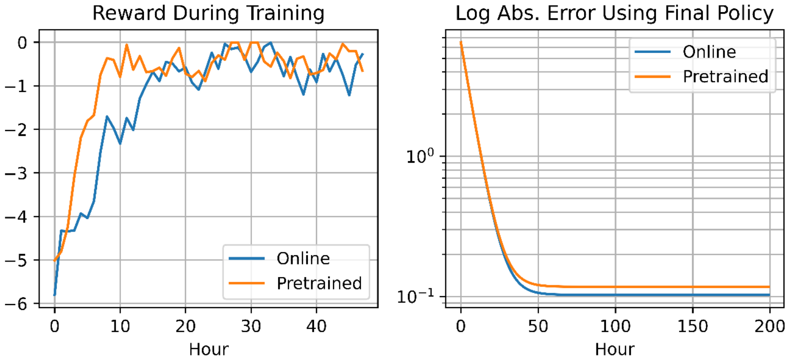

Figure 1: Rewards achieved during the online adaptation (left) and the dynamics of the absolute error value during the last policy rollout (right).

It was interesting to note that after reaching the plateau, the spread of rewards around the target value was slightly greater for the pretrained agent, which can be explained by the fact that during the fine-grained adjustment of the value of the target variable, it was hindered by previous experience in contrast to the agent without pretraining. This effect became more pronounced with the increase in simulation pretraining episodes and gradually disappeared with further online training, although the pretrained agent always converged to small absolute error values faster and better coped with random perturbations. Given this fact, we can conclude that when training for the task of maintaining biomass concentration (or any other parameters) at a fixed level, it is not worth doing long environment rollouts so that the agent can focus more on overcoming disturbances and world-model mismatches then on mastering the specific environment. Also, if necessary, additional regularization can be used during the simulation pretraining stage, which was not investigated in this work.

Table 4
Performance Metrics of Online-Only and Pretrained RL-Controllers

Pretraining	No	Yes
MAE	1.38497	0.97132
MSE	4.40333	2.62941
ITAE	19.50145	15.07084
ITSE	28.66606	16.4545

5. Conclusion

This article reviews key challenges in developing autonomous bioreactor control systems, including high complexity, non-linearity, stochasticity, limited system observability, measurement errors, and insufficient data—factors that complicate accurate modeling and control. The novelty of the study lies in the introduction of a hybrid approach to RL-based smart controller development that can help resolve the abovementioned problems via the combination of memory-based RL, staged offline-to-online integration, and (Recurrent) Soft Actor-Critic algorithm. Approbation of the effectiveness of the proposed method was performed by applying an RSAC agent, pretrained on two approximate deterministic simulations, to control a simulation of a fed-batch baker's yeast bioreactor with a partially observable state. Experiments demonstrated the ability of the agent created this way to adapt to the real environment and bring the system to the desired state much faster (6 hours vs. 14 without pretraining).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Spier, L. Vandenberghe, A. Medeiros, C. Soccol, Application of different types of bioreactors in bioprocesses, Bioreactors (2011) 53–87.

[2] G. Regonesi, Bioreactors: A complete review, 2023. doi:10.13140/RG.2.2.11630.79685.

[3] F. Palladino, A. Schlogl, A. Jose, R. Rodrigues, D. Fabrino, I. Santos, C. Rosa, C. Mario, Bioreactors: applications and innovations for a sustainable and healthy future—a critical review, Appl. Sci. 14 (2024) 9346. doi:10.3390/app14209346.

[4] F. Raganati, A. Procentese, Special issue on "bioreactor system: design, modeling and continuous production process", Processes 10 (2022) 1936. doi:10.3390/pr10101936.

[5] A. Soni, R. Parker, Closed-Loop control of fed-batch bioreactors: A shrinking-horizon approach, Ind. Eng. Chem. Res. - IND ENG CHEM RES 43 (2004). doi:10.1021/ie030535b.

[6] P. López-Pérez, R. Aguilar-López, R. Femat, Control in bioengineering and bioprocessing: modeling, estimation and the use of soft sensors., 2020. doi:10.1002/9781119296317.

[7] Y. Ma, D. Noreña-Caro, A. Adams, T. Brentzel, J. Romagnoli, M. Benton, Machine-Learning-Based simulation and fed-batch control of cyanobacterial-phycocyanin production in plectonema by artificial neural network and deep reinforcement learning, Comput. Chem. Eng. 142 (2020) 107016. doi:10.1016/j.compchemeng.2020.107016.

[8] E. Bolmanis, K. Dubencovs, A. Suleiko, J. Vanags, Model predictive control—a stand out among competitors for fed-batch fermentation improvement, Fermentation 9 (2023) 206. doi:10.3390/fermentation9030206.

[9] M. Monteiro, C. Kontoravdi, Bioprocess control: A shift in methodology towards reinforcement learning, 2024, pp. 2851–2856. doi:10.1016/B978-0-443-28824-1.50476-2.

[10] T. Oh, Quantitative comparison of reinforcement learning and data-driven model predictive control for chemical and biological processes, Comput. Chem. Eng. 181 (2023) 108558. doi:10.1016/j.compchemeng.2023.108558.

[11] Y. Haeun, H. Byun, D. Han, J. Lee, Reinforcement learning for batch process control: Review and perspectives, Annu. Rev. Control 52 (2021). doi:10.1016/j.arcontrol.2021.10.006.

[12] R. S. Sutton and A. G. Barto, Reinforcement Learning, second edition: An Introduction. MIT Press, 2018.

[13] N. Nievas, A. Pagès-Bernaus, F. Bonada, L. Echeverria, X. Domingo Albin, Reinforcement learning for autonomous process control in industry 4.0: advantages and challenges, Appl. Artif. Intell. 38 (2024). doi:10.1080/08839514.2024.2383101.

[14] N. Treloar, A. Fedorec, B. Ingalls, C. Barnes, Deep reinforcement learning for the control of microbial co-cultures in bioreactors, PLOS Comput. Biol. 16 (2020) e1007783. doi:10.1371/journal.pcbi.1007783.

[15] R. Sekhar, T. Radhakrishnan, S. Naina Mohamed, Deep deterministic policy gradient reinforcement learning based temperature control of a fermentation bioreactor for ethanol production, J. Indian Chem. Soc. 102 (2025) 101575. doi:10.1016/j.jics.2025.101575.

[16] R. Sekhar, T. Radhakrishnan, S. Naina Mohamed, Reinforcement learning based temperature control of a fermentation bioreactor for ethanol production, Biotechnol. Bioeng. 121 (2024) 3114–3127. doi:10.1002/bit.28784.

[17] Z. Klawikowska, M. Grochowski, Optimizing control of wastewater treatment plant with reinforcement learning: technical evaluation of twin-delayed deep deterministic policy gradient agent, IEEE Access PP (2024) 1–1. doi:10.1109/ACCESS.2024.3458186.

[18] H. Croll, K. Ikuma, S. Ong, S. Sarkar, Systematic performance evaluation of reinforcement learning algorithms applied to wastewater treatment control optimization, Environ. Sci. Technol. 57 (2023). doi:10.1021/acs.est.3c00353.

[19] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al., Soft actor-critic algorithms and applications, 2018. doi:10.48550/arXiv.1812.05905.

[20] W. Zhu, I. Castillo, Z. Wang, R. Rendall, L. Chiang, P. Hayot, J. Romagnoli, Benchmark study of reinforcement learning in controlling and optimizing batch processes, J. Adv. Manuf. Process. 4 (2022). doi:10.1002/amp2.10113.

[21] M. Biemann, F. Scheller, X. Liu, L. Huang, Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control, Appl. Energy 298 (2021) 117164. doi:10.1016/j.apenergy.2021.117164.

[22] Z. Xie, Z. Lin, J. Li, S. Li, D. Ye, Pretraining in deep reinforcement learning: A survey, 2022. doi:10.48550/arXiv.2211.03959.

[23] J. Pandian, M. Noel, Control of a bioreactor using a new partially supervised reinforcement learning algorithm, J. Process Control 69 (2018) Pages 16–29. doi:10.1016/j.jprocont.2018.07.013.

[24] P. Petsagkourakis, E. Bradford, D. Zhang, E. del Rio-Chanona, Reinforcement learning for batch bioprocess optimization, Comput. Chem. Eng. 133 (2019) 106649. doi:10.1016/j.compchemeng.2019.106649.

[25] H. Li, T. Qiu, F. You, AI-based optimal control of fed-batch biopharmaceutical process leveraging deep reinforcement learning, Chem. Eng. Sci. 292 (2024) 119990. doi:10.1016/j.ces.2024.119990.

[26] C. Luis, A. Bottero, J. Vinogradska, F. Berkenkamp, J. Peters, Uncertainty representations in state-space layers for deep reinforcement learning under partial observability, 2024. doi:10.48550/arXiv.2409.16824.

[27] M. Weissenbacher, A. Borovykh, G. Rigas, Reinforcement learning of chaotic systems control in partially observable environments, Flow, Turbul. Combust. (2025) 1–22. doi:10.1007/s10494-024-00632-5.

[28] M. Lingheng, R. Gorbet, D. Kulic, Memory-based deep reinforcement learning for POMDPs, 2021. doi:10.1109/IROS51168.2021.9636140.

[29] Y. Yang, Y. Jiang, J. Chen, S. Li, Z. Gu, Y. Yin, Q. Zhang, K. Yu, Belief state actor-critic algorithm from separation principle for POMDP, 2023. doi:10.23919/ACC55779.2023.10155792.

[30] Z. Yang, H. Nguyen, Recurrent off-policy baselines for memory-based continuous control, 2021. doi:10.48550/arXiv.2110.12628.