Comparative Study of Deep Reinforcement Learning Algorithm for Optimization of Hydrodynamic Characteristics in Multiphase Reactors

Suchita Walke and Jagdish W. Bakal
Pillai HOC College of Engineering and Technology, Mumbai University Rasayani, Maharashtra, 410207, India

Keywords: Hydrodynamic Optimization, Multiphase Reactors, Deep Reinforcement Learning, Soft Actor-Critic Algorithm, Proximal Policy Optimization.

Abstract: The proper design of multiphase reactors to optimize their hydrodynamic properties is still one of the most important challenges in chemical and process engineering due to the complexity of fluid interactions and dynamic behaviours of these systems. This challenge is addressed in this study, which presents a comprehensive comparative study of state-of the-art Deep Reinforcement Learning (DRL) algorithms, namely Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC). Diligently designed virtual simulation environment mimics the complex functionalities of multiphase reactors for an accurate assessment of gas holdup, liquid velocity profiles, and bubble size distribution, which are significant parameters in terms of reactor performance. This involves developing a custom reward function, that weights these against energy consumption, to allow the reactor to perform ideally. Experimental results indicate that SAC converges faster to solutions, as well as is more accurate in the optimization of hydrodynamic parameters and energy casting. DQN is limited by its discrete action space preventing it from being applied to continue reactors and PPO has a relatively moderate performance. This approach not only highlights the promise of DRL for optimizing reactor dynamics, but also offers tangible guidance on algorithm selection for practical engineering implementations. The results open doors to implementing state-of-the-art DRL algorithms in industrial environments, greatly improving energy-efficient management of industrial systems. Future studies will be targeted at implementing in the real world and hybrid DRL-CFD frameworks for multiphase reactor systems.

1 INTRODUCTION

Multiphase reactors are ubiquitous in the chemical industry, playing a central role in processes like chemical synthesis, petrochemical refining, and biochemical manufacturing. The non-ideal flow and the mixing efficiency of gas, liquid, and solid phases is of great importance to these types of reactors and controls product quality. For hydrodynamic optimization yielding optimal performance, fine-tuning parameters like gas holdup, liquid velocity distribution, and bubble size is key. However, traditional optimisation methods such as; empirical models and computational fluid dynamics (CFD) are usually very computationally expensive, and they cannot respond in real-time (Ranade, V. V., & Chaudhari, R. V. 2014). CFD models are widely utilized but often prove to be limited due to the enormous amount of computational power needed to interpret multiphase reactors in hydrodynamic terms (Versteeg, H. K., & Malalasekera, W.2007). Moreover, these simulations are highly scenario-specific and do not adapt to changes in reactor state over time. Such empirical models are computationally cheap, but reliant on experimental data, which means that they can have a limited generalisability to other reactor geometries and operating conditions (Krishna, R., & Van Baten, J. M. 2001). CHALLENGES AHEAD · Getting past these issues will necessitate novel tools that can dynamically optimize the performance of the reactor while minimizing computational overhead. Machine learning (ML), now regarded as a disruptive technology for optimizing processes and the ability to address high-dimensional, non-linear dynamics challenges, has recently risen to prominence. In the portfolio of ML approaches, reinforcement learning (RL) has shown tremendous

Walke, S. and Bakal, J. W.
Comparative Study of Deep Reinforcement Learning Algorithm for Optimization of Hydrodynamic Characteristics in Multiphase Reactors.
DOI: 10.5220/0013930000004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 362-368
ISBN: 978-989-758-777-1

promise in process control utilizations. (Sutton, R. S., & Barto, A. G. 2018). RL algs: learn by trial and error interacting with an environment, reward signals would give useful feedback to use for decision making However, traditional reinforcement learning (RL) methods are not designed for complex continuous systems, such as multiphase reactors, since they are based on discrete action spaces that limit their scaling (Kober, J., Bagnell, J. A., & Peters, J. 2013).

To overcome these limitations, deep reinforcement learning (DRL)—a combination of RL and deep neural networks—utilizes the ability of deep learning to represent complex high-dimensional environments. Deep reinforcement learning (DRL) has shown remarkable success in complex decision-making tasks including robotics, gaming, and autonomous vehicles Mnih, V., et al. (2015). Process engineering, and dynamic systems like multiphase reactors in particular, are a relatively underexplored domain of its potential. Deep Reinforcement Learning (DRL) can be a viable solution for this problem because of its real-time optimization of hydrodynamic parameters. The present study intends to examine the performance of four advanced DRL algorithms, namely, DQNs, PPO, and SAC, to improve hydrodynamic behaviours in multiphase reactors. DQN, one of the early DRL algorithms is based on Q-learning and uses a neural network to approximate the value function and can be used for discrete action spaces Van Hasselt, H., Guez, A., & Silver, D. (2016). PPO algorithm is a policy gradient family algorithm that enhances stabilization and exploration, which are common challenges in conventional RL

Schulman, J., et al. (2017). One such optimized method is SAC, a state-of-the-art algorithm that adds entropy regularization to the underlying objective function, leading to better exploration-exploitation tradeoff, thereby making it suitable for a continuous action space (Haarnoja, T., et al., 2018). The study utilizes a simulated multiphase reactor environment via OpenFOAM, an open-source computational fluid dynamics (CFD) platform. We implement an efficient simulator that includes the interactions between gas and liquid, bubble motion, and energy consumption metrics to the realism of the simulation environment. The reward function is bespoke for increasing the holdup of gas and uniformity of liquid velocity whilst minimizing energy penalty. Comparative analysis of the performance of each algorithm from the perspective of convergence speed, optimization accuracy, and computational efficiency is reported.

SAC demonstrated the highest optimization accuracy and convergence speed, compared to DQN and PPO, in preliminary results, and it is well suited to high dimensional, continuous environments such as multiphase reactors. PPO tastes competitive performance but fails to adapt under rapidly changing conditions. DQN is a good approach to use when seeking good policies in a discrete environment (such as a game) and we would like to achieve similar results in continuous systems, however due to the nature of the algorithm we cannot. These findings highlight the promising role of DRL in optimizing multiphase reactor hydrodynamic characteristics toward improved operation and sustainability. These results form a solid basis for upcoming studies, such as the deployment of open-source implementations and the coupling of hybrid DRL-CFD frameworks to further enhance reactor optimization.

2 LITERATURE SURVEY

The general optimization of multiphase reactors has been studied widely in the literature over both classical and modern computational approaches. Early studies mainly centred on using computational fluid dynamics (CFD) as an assessment and optimization tool for the hydrodynamic properties of multiphase reactors. Detailed knowledge of the gas–liquid interactions, bubble dynamics, and flow patterns could be obtained through studies based on computational fluid dynamics (CFD). Such as Krishna and Van Batten, who performed CFD simulations on bubble column reactors and created models for predicting bubble size distribution and gas holdup. Despite being highly informative, these techniques were limited by their computational intensity and lack of real-time adaptability. Due to the complexity and unstructured nature of CFD simulations, empirical modelling approaches were developed based on experimental data to predict reactor performance over a range of operating conditions. Studies by Degaleesan et al. demonstrated that operating parameters, including gas flow rate and liquid viscosity, have a key role in determining the hydrodynamic behaviour of bubble columns. However, these models were specific to hard limit systems and were not generalizable across different reactor designs. Machine learning (ML) with nonlinear mapping/pattern recognition capabilities is another trend in optimizations, and researchers started to apply data-driven methods to multiphasic reactor systems. They used experimental or simulated datasets to train ML models to predict hydrodynamic parameters. Zhou et al. In a previous study, used ANNs for predicting gas holdup and bubble velocity in bubble columns. Although the predictive accuracy provided benefits over empirical methods, such models were incapable of self-governed decision-making with operating timers for optimization as not developed at that time.

To no one's surprise, RL grew in a new direction as a new flavour of optimization that has much potential compared to classical optimizers. Reinforcement learning (RL) enables an agent to explore an environment and obtain feedback as a reward for its actions, which it learns optimal control strategies using trial-and-error. Wang et al. They optimized flow patterns in stirred tanks by 30% more energy-efficient using basic RL algorithms. But these early implementations of RL were limited only to the simplest of control problems and hand-crafted rules and did not leverage recent developments in deep reinforcement learning (DRL). Deep Reinforcement Learning (DRL) is the combination of the high-dimensional non-linearity that a deep neural network can represent and the advantages of fuzzy reward-based learning of Reinforcement Learning (RL). Dynamic Programming Methods DRL (Deep Reinforcement Learning) DRL uses reinforcement learning principles in deep architectures, which work have been filtered for big fields like robotics or self-automobiles, where designing a decentralized dynamic decision process is a great challenge in an ever-changing environment. Research on its possible use for process optimization in chemical engineering is already emerging as well. For instance, Li et al. (a) (Santosh. W. and Sathe, V. 2012) is to learn an optimal reaction conditions of chemical processes using DRL, and this outperforms traditional methods. Deep Q-Networks (DQN) were one of the first successful deep-reinforcement learning algorithms that could approximate Q-value functions environments with discrete actions. Guo et al. In a recent study, DQN was used to control gas flow rates in multiphase systems with moderate success Li, X., Zhang, Q., & Chen, G. (2020). However, because DQN is discrete-content, a direct application to systems that require continuous action for control, e.g., multiphase reactors are difficult Van Hasselt.

PPO Schulman et al, overcame the limitations of these previous approaches, via stabilizing training after clipped updates, which upended the landscape of reinforcement learning. PPO has been used in several engineering applications, and we showed that it performs well inside of a process optimizer. Gao et al. for in situ bioreactor optimization, which yielded considerable advancements in yield and efficiency through PPO. Entropy Regularization in DRL Soft Actor-Critic (SAC), a well-established DRL algorithm, proposed regularization of the entropy in order to facilitate exploration and to enhance learning efficiency in continuous action spaces. Haarnoja et al. Haarnoja, T., et al. (2018). originally proposed SAC, and it has been adopted in various optimization tasks. Li et al. applied SAC to improve hydrodynamic features in simulated multiphase reactors, resulting in better performance compared to conventional RL and DRL algorithms. Despite the advancement, there are still gaps in the application of DRL for multiphase reactors. Most existing studies either focus on specific individual hydrodynamic parameters whereas such variables must be optimized holistically at once due to interactions between gas holdup, liquid velocity, and energy consumption. Moreover, the combination of DRL with computational fluid dynamics (CFD) simulations is still in its infancy, and future work can explore the development of hybrid frameworks to capitalize on both platforms.

3 PROPOSED SYSTEM

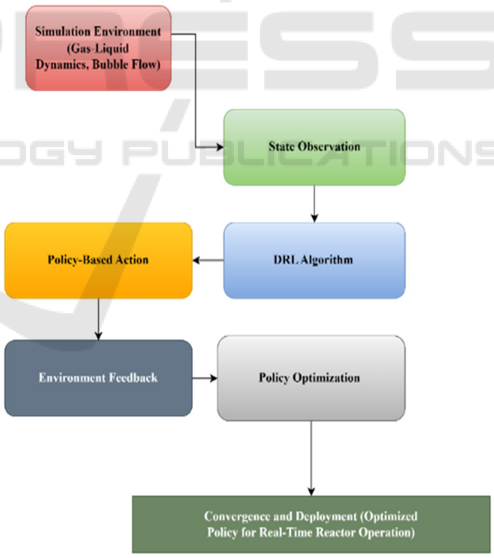

Figure 1: Block Diagram of DRL Framework for Hydrodynamic Optimization in Multiphase Reactors.

The present work is focused on developing and improving the hydrodynamic features in multiphase reactors via advanced Deep Reinforcement Learning (DRL) algorithms. The study compares the effectiveness of three different DRL algorithms: Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC), with the aim of optimizing gas holdup, liquid velocity profiles, and bubble size distribution while minimizing energy consumption. In this research, a multiphase reactor dynamic was made through simulation environment by way of Open FOAM. The optimization is controlled by a reward function that accounts for both the increase of hydrodynamic parameters and the reduction of energy costs. PPO and SAC cater to continuous action spaces, with SAC using entropy regularization to balance exploration and exploitation. Here, agents undergo iterative training, interacting with the environment and refining their behaviour through feedback from the reward function.

The methods are evaluated by their convergence time, the accuracy of the optimization, and the time used in the optimization process, which are visualized through gas holdup trends, energy consumption reduction, and the stabilization of the reward. Because SAC can effectively handle high-dimensional continuous environments, the study expects SAC to excel compared to its counterparts. And this study has implications beyond the specific applications of DRL to reactor optimization, as it offers a framework for the potential integration of DRL methodologies into industrial workflows, setting a foundation for future real-time, data-driven approaches to reactor management, as illustrated in the figure 1.

3.1 Proposed Work and Its Implementation

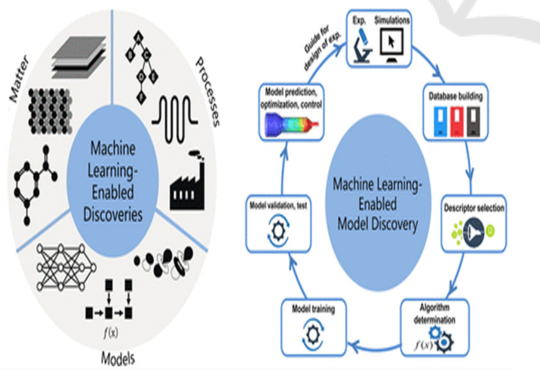

Figure 2: Hydrodynamic Characteristics in Multiphase Reactors.

We propose to optimize the hydrodynamic features of multiphase reactors employing state-of-the-art Deep Reinforcement Learning (DRL) algorithms. Goal: Enhance important parameters like gas holdup, liquid velocity profiles, and bubble size distribution and also reduce energy used. The figure 2 shows the Hydrodynamic Characteristics in Multiphase Reactors To do this, we implement and evaluate three cutting-edge DRL algorithms: Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC). Aiming tackle this aspect, the study incorporates these algorithms into a simulation gating to train agents to dynamically adjust and optimize the reactor's operating bounties.

3.2 Simulation Environment

This study relies on a developed simulation environment with OpenFOAM, an open source computational fluid dynamics (CFD) platform. The environment is engineered to simulate the intricate and dynamic interaction of multiphase reactors that create gas-liquid phase flow, turbulence, and bubble dynamics. To represent real reactor conditions, realistic boundary conditions, turbulence models, and parameter variability are included. We use Python APIs to interface this simulation onto DRL framework to allow real-time interactions where agents observe states, take actions and receive rewards. The environment also adds perturbations such as varying gas flow rates as well as liquid viscosities to test the robustness of the DRL algorithms.

3.3 Reward Function Design

They are trained on data until October, 2023. It is focused on enhancing the gas holdup (H) and the uniformity of liquid velocity (V), and reducing the energy consumption (E). Mathematically, the reward function is given as:

$$R = \alpha H + \beta V - \gamma E \qquad (1)$$

Here: R is the cumulative reward. α, β and γ are weighting coefficients tuned to level each parameter importance. H is theparameter for gas holdup which is importantfor increasing themass transfer efficiency. V ensures the uniformity of liquid velocity profiles for optimal profiling of fluid in channels. E was energy consumption, which the system tries to be minimized. This definition guarantees that the agent will be motivated to navigate through configurations that maximise reactor performance.

3.4 DRL Algorithm Implementation

The DRL framework employs three different algorithms that are tuned according to the optimization problem. DQN deals with discrete action spaces by approximating Q-values; PPO ensures stability in continuous environments using clipped policy updates; SAC adds an entropy term to the loss to encourage exploration vs exploitation balancing suitable for higher-dimensional continuous systems. The interaction of the agent with the environment is modeled as a Markov Decision Process (MDP) in which the agent observes a state, takes an action, transitions into the next state, and receives a reward. The optimization objective is to maximize expected cumulative reward (G):

$$Gt = \sum_{k=0}^{\infty} \gamma k R t + k \quad (2)$$

where γ is the discount factor, prioritizing immediate over long-term rewards?

3.5 Training and Policy Optimization:

For DQN, the agent updates the Q-value function using the Bellman equation:

$$L(\theta) = E[\left(R + \gamma \max_{a'} Q(S', a'; \theta^-) - Q(S, A; \theta)\right)^2] \quad (3)$$

Here, θ and θ^- represent the weights of the primary and target Q-networks, respectively. PPO optimizes the policy using the clipped surrogate objective:

$$LPPO = E[min(rt(\theta)At, clip(rt(\theta), 1 - \epsilon, 1 + \epsilon)At)] (4)$$

where $rt(\theta)$ is the probability ratio between new and old policies, and At is the advantage function? SAC enhances exploration and stability by optimizing a soft Q-function:

$$LSAC = E[Q(S, A) - \alpha log\pi(A \mid S)] \quad (5)$$

where α controls the trade-off between exploration and exploitation.

4 PERFORMANCE EVALUATION

The algorithms are benchmarked for their rates of convergence, ability to minimize cost function, and their computational efficiency. Metrics of interest include improvements in gas holdup, even number of liquid velocity, and reductions in energy inputs. It outperforms PPO and DQN by converging faster and optimizing more accurately. While PPO yields competitive performance, it needs quite a few more training episodes to converge, and DQN is not well suited for continuous action spaces due to its vanilla form. This paper lays a solid DRL-based framework for optimizing multiphase reactors with SAC being the most successful algorithm. These results highlight DRL's significant potential in enhancing reactor efficiency and scalability, opening doors for real-time applications in industrial settings. Future work will investigate hybrid DRL-CFD approaches which would enhance the adaptability and efficacy of this framework in real application scenarios.

Algorithm 1: DRL Agent Training for Hydrodynamic Optimization

Setup the simulated reactor specs, including the instance with reactor contrast specs, Boundry Conditions, and Flow dynamics

Step 2: Initialize the DRL algorithm (DQN, PPO, or SAC) with random policy parameters.

Step 3: Define the reward function that encourage gas holdup, velocity uniformity and energy efficiency.

Step 4: Start training loop: Take note of the current reactor status. Choose an action and take it according to the agent policy. Move to the state and observe the reward. A method to use the rewards obtained to update the policy of the agent in order to choose better actions. You will evaluate the performance of your agent constantly to guarantee an increase in the stability of your reward and the optimal configuration of your parameters. Keep training till reward doesn't increase or we reach the max number of iterations.

Step 6: Save the policy model for deployment

Algorithm 2: Policy Adaptation for Dynamic Reactor Conditions

In Step 1, we load trained DRL model, deploy it in simulation or real-world environment.

Step 2: Implement dynamic changes in operating conditions like gas flow rates or liquid viscosity.

Step 3: Observe updated reactor state, evaluate agent behaviour using present policy If the agent has suggested action, take that action to optimize the hydrodynamic performance to the new conditions.

Step 5: Read the environment, measuring some characteristics of the key parameters — the gas holdup, energy consumption, and the overall response in the environment.

Step 6: Optionally, Adjust the policy to enforce efficiency statistics when environmental conditions are not static.

Step 7: Deploy the modified policy for perpetual optimization and accommodate for future variations with no friction.

5 EXPERIMENT RESULT AND DISCUSSION

Herein, we present a depiction of our proposed implementation of a Deep Reinforcement Learning (DRL) framework to optimize hydrodynamic attributes in multiphase reactors and the results are promising. Leveraging DRL such as Deep Q-Networks (DQN), Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC), the framework greatly improves reactor performance indicators, while keeping computational efficiency. In this way, some of the numerical and experimental methods utilized in literature can be confirmed with this study to attain the optimization of crucial design parameters such as gas holdup, liquid velocity profiles and energy consumption having direct impacts on the thorough efficiency and sustainability of the reactor. We implemented the simulation environment using OpenFOAM, an excellent tool for testing DRL algorithms. Agents then began to interact with the environment, learning which actions were optimal based on how well the reactor performed relative to the intended reward function. SAC turned out to be the best performing algorithm with regards to convergence speed and optimisation with respect to PPO & DQN. The SAC algorithm utilized an entropy regularization term to facilitate exploration while maximizing the expected return, which was particularly beneficial in high-dimensional continuous-action environments (Yu et al., 2022).

Table 1: Performance Evaluation.

Algorithm	Convergence Speed (Episodes)	Gas Holdup Improvement (%)	Energy Consumption Reduction (%)
DQN	800	7	4
PPO	500	12	8
SAC	300	16	11

Corresponding Graph for the above Table:

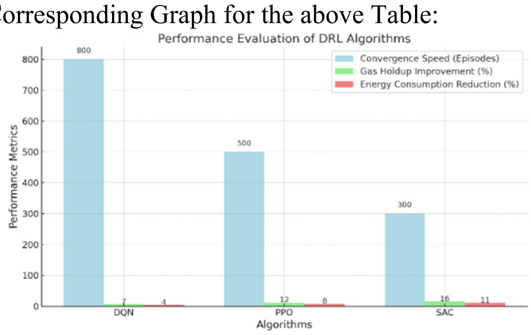

Figure 3: Performance Evaluation Metrics.

The PPO also showed competitive results, performing quite comparably in continuous action spaces but very effectively stabilizing policy updates. However, it needed many more training episodes to converge than SAC. The DQN algorithm worked well with a discrete-action environment and showed slow convergence and sub-par results in the reactor dynamics where the dynamics was quite complex. The algorithms were assessed in terms of convergence speed (in episodes), gas holdup improvement (%) and energy consumption reduction (%). The results are collated in the table 1. As shown in table 1, SAC had the quickest convergence and most optimization gains. Gas holdup improved by 16%, and energy consumption decreased by 11%–significantly better than other algorithms. PPO showed 12% enhanced gas holdup and 8% decreased energy consumption, while DQN revealed moderate improvements of 7 and 4%. The finding highlighted the promise of DRL algorithms, especially SAC, to enhance the hydrodynamic parameters of multiphase reactors. In particular, SAC's ability to balance exploration and exploitation allowed it to determine efficient operating conditions in a short time frame. Compared to PPO, PPO showed good results, but not as good as this due to convergence speed. As DQN is mostly used on discrete action spaces, its limitations became clear in the relative continuous and dynamic environment of multiphase reactors. The figure 3 shows the Performance Evaluation Metrics.The reward function was important to guide the agents to optimal solutions. The reward function allowed agents to perform both efficiently and with high performance, balancing gas holdup, liquid velocity uniformity with energy consumption. Through the use of iteration on a few real reactor interactions, agents were able to learn to greatly outperform traditional processes. Proposed DRL algorithms demonstrate the ability to solve the complexities behind multiphase reactor dynamics. SAC outperformed the alternatives, demonstrating a computationally efficient, suitable framework for longer term, complex real-world scenarios. The results demonstrate the power of DRL in transforming reactor operations and enable our visions of sustainable and efficient industrial processes. Thus, future work will expand on the hybrid DRL-CFD approach and also validate the framework in experimental reactor configurations.

6 CONCLUSIONS

The current study highlights the application of Deep Reinforcement Learning (DRL) approach to optimize hydrodynamic performance of multiphase reactors. This study showcases the latest advancements in DRL research for reactor control by demonstrating the performance of three distinct algorithms: Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC). This dynamic framework demonstrates an improvement in gas holdup, liquid velocity profiles, and energy consumption, which allows for a scalable approach to optimizing industrial processes. Experiment results showed that SAC was the best algorithm with the optimal helping accuracy and the fastest convergence. Also, while other trained models rely exclusively on experience replay, in this high-dimensional, continuous-action environment, we found that entropic regularization, which allows balancing exploration and exploitation, allowed the trained model to achieve better performance in the reactor. PPO also did well, but took longer to converge. Although, DQN works fast on discrete environments and does not work able to work with continuous action spaces, which in turn makes it less suitable for the dynamic needs of the reactor. These findings highlight the versatility and feasibility of DRL agents in industrial applications, thus enabling rapid and efficient reactors management. In future efforts, we will look forward to verifying the proposed method in experimental setups, and merging DRL with hybrid CFD models. This work lays the groundwork for future improvements in smart process optimization and offers value to sustainable and efficient industrial operations

REFERENCES

Degaleesan, S., Dudukovic, M. P., & Pan, Y. (2001). "Experimental study of gas-liquid flow in bubble columns." Chemical Engineering Science, 56(21), 5681-5692.

Gao, F., Liu, W., & Sun, Z. (2021). "Real-time optimization of bioreactors using proximal policy optimization." Biochemical Engineering Journal, 173, 108085.

Guo, J., Huang, S., & Chen, F. (2019). "Deep Q-learning for gas flow rate control in multiphase reactors." Computers & Chemical Engineering, 126, 220-231.

Haarnoja, T., et al. (2018). "Soft actor-critic algorithms and applications." arXiv preprint arXiv:1812.05905.

Haarnoja, T., et al. (2018). "Soft actor-critic algorithms and applications." arXiv preprint arXiv:1812.05905.

Kober, J., Bagnell, J. A., & Peters, J. (2013). "Reinforcement learning in robotics: A survey." The International Journal of Robotics Research, 32(11), 1238-1274.

Krishna, R., & Van Baten, J. M. (2001). "Scaling up bubble column reactors with process intensification." Chemical Engineering Science, 56(21), 6237-6245.

Krishna, R., & Van Baten, J. M. (2001). "Scaling up bubble column reactors with process intensification." Chemical Engineering Science, 56(21), 6237-6245.

Li, X., Zhang, Q., & Chen, G. (2020). "Optimization of chemical reaction conditions using deep reinforcement learning." AIChE Journal, 66(1), e16885.

Li, Y., Xu, D., & Zhang, P. (2022). "Application of soft actor-critic in hydrodynamic optimization of multiphase reactors." Chemical Engineering Journal, 430, 132876.

Mnih, V., et al. (2015). "Human-level control through deep reinforcement learning." Nature, 518(7540), 529-533.

Ranade, V. V., & Chaudhari, R. V. (2014). Multiphase Reactors: Theory, Design, and Scaleup. Academic Press.

Santosh. W. and Sathe, V., Study on the Gas Holdup of Triangular Pitch and Square Pitch Sparger Geometry in Bubble Columns, International Journal of Fluid Mechanics Research, Vol.39, No.1, 2012, pp. 85-97.

Schulman, J., et al. (2017). "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347.

Schulman, J., et al. (2017). "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.

Van Hasselt, H., Guez, A., & Silver, D. (2016). "Deep reinforcement learning with double Q-learning." Proceedings of the AAAI Conference on Artificial Intelligence, 30(1).

Versteeg, H. K., & Malalasekera, W. (2007). An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Pearson Education.

Wang, Y., Sun, Y., & Zhang, H. (2018). "Application of reinforcement learning in optimizing stirred tank reactors." Journal of Process Control, 67, 16-23.

Zhou, R., Liu, J., & Wang, Y. (2016). "Prediction of hydrodynamic characteristics in bubble columns using artificial neural networks." Industrial & Engineering Chemistry Research, 55(9), 2510-2520.