Journal of Green Chemical and Environmental Engineering

e-ISSN 3090-2568
p-ISSN 3109-2233

Review Paper

Reinforcement Learning for Intelligent Engineering Systems: A Comprehensive Review of Applications, Challenges and Future Prospects

Samuel Boluwatife Giwa, Aminah Abolore Sulayman*, Kazeem Kolapo Salam, Dauda Olurotimi Araromi

Chemical Engineering Department, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

** Corresponding author: aasulayman@lautech.edu.ng (Aminah Abolore Sulayman)*

Received 14 July 2025
Revised 25 August 2025
Accepted 30 Sept 2025

Citation: S. B. Giwa, A. A. Sulayman, K. K. Salam, and D. O. Araromi, (2025). "Reinforcement Learning for Intelligent Engineering Systems: A Comprehensive Review of Applications, Challenges and Future Prospects". J. of Green Chemical and Environmental Engineering, Vol. 1, No. 3, 184-201.

: 10.63288/jgcee.v1i3.15

Abstract: Reinforcement Learning (RL) is revolutionizing the field of engineering through the solution of challenging, nonlinear, and high-dimensional problems. This review examines how RL enriches the subjects of engineering, such as optimization of industrial processes. Current techniques in optimization and control are inefficient for some complex systems, but RL serves as a better alternative through real-time optimization, product quality improvement, and optimization of process efficiency. The article focuses on recent advancements, challenges, and future prospects for extended integration of RL in engineering and its possibility to revolutionize the field. It also states its limitations and suggestions for future research. The review serves as a good source of information for researchers and engineers determined to remain up to date with recent advancements in RL for intelligent engineering systems and extend its development.

Keywords: Reinforcement Learning; engineering; artificial intelligence; algorithms; complex system.

1. Introduction

The increasing development of intelligent automation has put Reinforcement Learning (RL) at the forefront of innovation in various engineering applications [1] . RL, a type of machine learning algorithm, uses an agent to make decisions in a dynamic environment where the agent learns by feedback from interactions with the environment [2]. RL's ability to operate effectively in complex, uncertain, and nonlinear systems makes it an attractive option for solving problems in engineering areas such as process control and optimization, process design, and fault diagnosis [3]. Unlike supervised learning, which is trained on a labeled dataset, RL's interaction with the environment enables it to learn optimal behaviors via trial-and-error, which are reinforced by the reward signals received from the interaction [4]. Additionally, RL's adaptability to real-time changes is advantageous for complex engineering problems, which range from robotics, autonomous systems to industrial process optimization and energy management [5].

RL goal is to address limitations of traditional supervised learning, which relies on labeled data provided by a supervisor[2]. Unlike supervised models that cannot go beyond the information contained in the training set, RL agents learn an optimal policy through interaction with the environment, using feedback in the form of rewards to improve performance[6]. This policy which

maps state to action is learnt through trial-and-error search guided by a scalar reward signal. This guided search introduces unique challenges relevant to engineering problems, particularly the trade-off between exploration and exploitation. The agent seeks to maximize cumulative rewards, but it must explore the state space to identify effective actions. In engineering control applications, this mirrors the well-known dilemma of identification (estimation) versus control: a controller must sometimes perturb a system to improve its model (exploration) before applying actions that maximize performance (exploitation). For stochastic systems such as chemical reactors, energy systems, or manufacturing processes, reliable decision-making requires repeated exploration of states to reduce uncertainty. When rewards are delayed as in thermal systems where the effect of a control action may only be observed minutes later [7]. This makes identifying optimal strategies even more complex. Moreover, in non-stationary or time-varying systems, which are common in real-world engineering (e.g., fluctuating feed conditions in chemical processes), continuous exploration is necessary to maintain near-optimal control policies [8].

Furthermore, RL meets the demand of modern engineering operations by offering a flexible framework that balances exploration of new strategies with the exploitation of learned knowledge, enabling learning even in partially known environments [9]. This might not be the case for traditional control strategies as they might fail in certain scenarios. This scenario includes the absence of accurate mathematical models, model uncertainties, plant-model mismatch, and nonlinearities. The high computational cost of system identification techniques, as well as the time-varying behavior of dynamical systems, might also complicate traditional control strategies. Advances in computational power and algorithmic development have accelerated the adoption of RL techniques. Early approaches were based on tabular RL, such as Q-learning and SARSA, where value functions are stored explicitly in a Q-table[10]. While the tabular approach is effective for small, discrete environments, it becomes cumbersome as state-action spaces grow, due to the curse of dimensionality[11]. To overcome this limitation, researchers introduced function approximation techniques, paving the way for Deep Reinforcement Learning (DRL). These methods include Deep Q-Networks (DQN), Policy Gradient Methods (PGM), Actor-Critic Architectures (ACA), and Model-Based RL (MBRL) [12]. More recent Deep Reinforcement Learning (DRL) algorithms include Proximal Policy Optimization (PPO), which stabilizes training by clipping policy updates[13]. Deep Deterministic Policy Gradient (DDPG) is efficient in handling continuous action, and Soft Actor-Critic (SAC), leverages entropy maximization to encourage exploration while achieving superior performance in high-dimensional tasks[14][15]. RL algorithms have shown remarkable success within engineering domains, achieving near-optimal control in complex scenarios. Moreover, hybrid frameworks that combine RL with classical control methods, such as Model Predictive Control (MPC), or incorporate physics-based models, offer potential pathways for enhancing safety, robustness, and interpretability [16].

There are several surveys on reinforcement learning, however, most are restricted to algorithmic development or narrow application domains. Some studies are specific to a concept in process industries, such as process control, which omits other engineering domains. However, there is less emphasis on cross-domain challenges or unified frameworks. For instance, a review by Dogru et al., [17] is specifically on process control, while Weinberg et al. [18], is specifically on RL maintenance. Hence, the objective of this paper is to highlight and showcase the potential of RL. By bridging theory with practice, this review not only explains the mathematical basics of RL algorithms but also includes case studies and real-world examples that show their effectiveness. Additionally, it highlights ongoing challenges, including sample inefficiency, safety limitations, and interpretability, providing insights into potential new solutions and future directions. This overview also acts as a

fundamental resource for researchers and students seeking to understand and use RL in engineering fields. The review does not sufficiently address the maturity and readiness of RL methods for industrial use. Issues such as interpretability, and integration with existing control systems are mentioned but not deeply analyzed from a practical adoption perspective. In lieu of these, this review focuses on the comprehensive integration of reinforcement learning research across diverse intelligent engineering systems, its critical examination of application-specific challenges (safety, interpretability, stability, and data limitations), and its forward-looking perspective that outlines future research prospects tailored to engineering practice.

2. Fundamentals of reinforcement learning

RL is based on Markov Decision Processes (MDPs), which is a mathematical framework used to describe decision-making problems in environments where outcomes are stochastic and partly under the control of an agent, which is the chief decision maker [19]. The MDP is a five-parameter turple that includes the State(S) space, Action (A) space, Reward (R) Function, policy (π), and discount factor (γ). An agent is the decision-maker in reinforcement learning. A typical RL episode is present in Figure 1, at each time step, the agent observes the current state of the environment. Based on this state, the agent chooses an action according to its policy (the behavior). The environment then responds to this action by transitioning to a new state and producing a reward, which is a numerical signal that quantifies the consequence of the action.

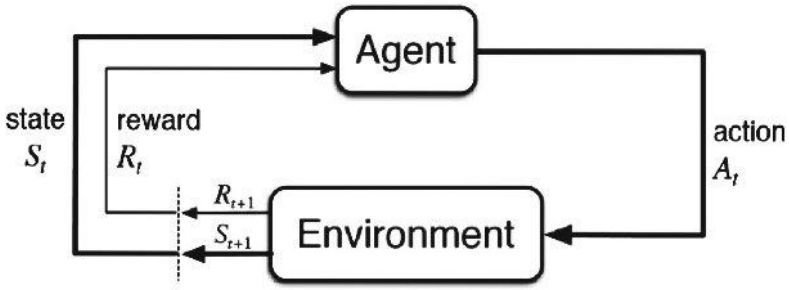

Figure 1. RL diagram [20]

The agent's goal is not just to maximize the immediate reward (short-term feedback), but to maximize the expected return (which captures future rewards). The expected return, G_t at time step t is expressed in Equation 1.

$$G_t = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}\right) \quad (1)$$

$\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$ is an infinite discounted sum of future rewards. γ ∈ [0,1) is the discount factor, γ= 0 favors immediate reward, γ close to 1 the agent values long-term rewards almost equally to immediate ones. $\mathbb{E}_\pi(.)$ is the expectation under policy, π. The policy governs the agent's behavior. r_t is the reward received at time t, and k represents the number of steps into the future. The agent iteratively updates its policy to maximize this return through interaction with the environment.

3. Classification of Reinforcement Learning

RL algorithms can be classified based on different criteria that reflect their interaction and learning strategies with the environment. They can be categorized into either model-free or model-based approaches based on whether an explicit model of the environment is used. In a model-free approach, the agent does not require knowledge of the dynamics of the environment, such as the state transition probabilities or the reward model [21]. The agent learns from experience through trial

and error, ultimately determining the best actions based on the rewards it receives after taking each action. Conversely, model-based RL involves obtaining the optimal behavior by training a model that encapsulates the environment's dynamics [22]. This model helps in predicting the next state and reward given the current state and action. Further classifications of RL algorithms are based on Policy type, Action space, learning paradigm, policy interaction, and value estimation method. These classifications provide a structured understanding of RL approaches, with their descriptions presented in Table 1.

Table 1. RL classification

	Category	Description	Examples	References
Model Usage	Model-Free	Learns from experience without modeling environment dynamics.	Q-Learning, SARSA, REINFORCE	[21]
	Model-Based	Learns or uses a model of the environment to simulate transitions and rewards.	Dyna-Q, PETS, MuZero	[22]
Policy Type	Value-Based	Learns value functions (e.g., Q-values); chooses actions indirectly.	Q-Learning, Deep Q-Learning (DQN)	[23]
	Policy-Based	Learns the policy directly; suitable for continuous action spaces.	REINFORCE, Proximal Policy Optimization (PPO)	[24]–[26]
	Actor-Critic	Simultaneously learns both policy and value function.	Advantage Actor-Critic (A2C) ,Deep Deterministic Policy Gradient(DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3)	[25], [26]
Action Space	Discrete	Agent chooses from a finite set of possible actions.	Gridworld, CartPole	[27]
	Continuous	Agent chooses from an infinite set of actions.	MuJoCo, Robotics Tasks	[27]
Learning Paradigm	Online	Agent interacts with the environment while learning.	DQN, PPO, A2C	[28], [29]
	Offline	Agent learns from a fixed dataset without further environment interaction.	Batch RL, Conservative Q-Learning (CQL)	[28][30]
Policy Interaction	On-Policy	Learns from data collected using the same policy it's trying to optimize.	State-Action- Reward-State-Action(SARSA), A2C	[30]
	Off-Policy	Learns from data collected using a different behavior policy.	Q-Learning, DDPG, TD3	
Value Estimation Method	Dynamic Programming	Requires a complete model of the environment (transition probabilities & rewards).	Value Iteration, Policy Iteration	[31]–[33]

Category	Description	Examples	References
Monte Carlo(MC)	Does not require a model; learns from complete episodes.	First-Visit MC, Every-Visit MC	
Temporal Difference (TD)	Model-free; updates value estimates from partial episodes.	TD(0), SARSA, Q-Learning	
TD(λ) & n-step	Generalized TD methods handling both episodic and continuous tasks.	TD(λ), n-step Q-Learning	

4. Algorithms in Reinforcement Learning

4.1. State-Action-Reward-State-Action (SARSA) and Q learning

State-Action-Reward-State-Action (SARSA) and Q-learning are both value-based reinforcement learning algorithms designed to find optimal policies by estimating action-value functions, commonly called Q-values [22, 23]. The primary difference between them lies in how they update these values [35]. Q-learning is an off-policy method that assumes the agent always takes the optimal next action, using the maximum Q-value of the next state in its updates [36]. In contrast, SARSA is an on-policy method that updates based on the action the agent takes, considering exploration strategies like ε-greedy [37].

Both algorithms utilize a Q-table to store expected rewards for state-action pairs and depend on trial-and-error interactions with the environment to improve their decision-making over time. While SARSA updates reflect the current policy being followed, Q-learning converges toward the optimal policy if there is sufficient exploration. Q-learning has several advanced variants, including Deep Q-Learning (which uses neural networks for function approximation), Hierarchical Q-Learning (which breaks problems into sub-tasks), and Nash Q-learning (used in multi-agent environments) [38]–[41]. These variants have been successfully applied in domains such as energy management, autonomous navigation, load balancing in power grids, and optimization tasks like PV array reconfiguration with hydrogen energy storage [42].

4.2. Policy-Based Algorithms

Policy-based reinforcement learning algorithms directly optimize the policy without estimating value functions first [43], [44]. The work used Policy Gradient (PG) methods, which update the policy parameters via Stochastic Gradient Ascent. Algorithms in this class include REINFORCE, Actor-Critic, Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Proximal Policy Optimization (PPO). REINFORCE used Monte Carlo sampling to update policies based on full episode returns [45]. Actor-Critic (AC) combines value and policy learning, where the actor updates the policy and the critic evaluates actions. DDPG extends Q-learning to continuous spaces by using deterministic policies and target networks [46]–[48].TD3 refines DDPG with twin critics and delayed updates for improved stability [49]. PPO improves training by clipping large policy updates, enhancing

robustness, and performance [13], [50]. A detailed description of the various policy-based algorithm is presented in Table 2.

Table 2. Policy based algorithm

Algorithm	Type	Key Idea	Use Case
REINFORCE	Monte Carlo PG	Update policy after full episode using returns	Simple episodic tasks
Actor-Critic	Hybrid	Actor learns policy; the critic estimates value function	Stable policy learning
DDPG	Deterministic AC	Actor-critic for continuous control, uses target networks and Polyak averaging	Robotics, continuous spaces
TD3	Improved DDPG	Uses twin critics, noise, and delayed actor updates	Reduces overestimation, stable
PPO	Clipped PG	Restricts policy updates to avoid instability	Most stable and widely used

5. Applications of Reinforcement Learning (RL)

RL has gained significant attention as an advanced control strategy in real-time industrial systems, which often suffer from nonlinearities, disturbances, model uncertainties, and constrained inputs [51]. These issues, especially the uncertainty in control gain for affine nonlinear systems, pose difficulties for classical controllers like Proportional Integral Derivative (PID), Linear Quadratic Regulator (LQR), and Nonlinear Model Predictive Control (MPC) [52]–[55]. RL offers a model-free alternative, learning optimal control policies directly from interaction with the environment and enabling superior trajectory tracking without explicit system models [56]. Studies have applied RL in various domains from wheeled mobile robots using actor-critic structures [56] to chemical reactors using Monte Carlo-enhanced DDPG for economic and terminal constraint satisfaction [57]. Actor–critic methods offer exploration and policy richness; however, critic-only and hybrid approaches are more computationally efficient and interpretable.

The integration of RL with Model Predictive Control (MPC) has proven to accelerate setpoint tracking and automate tuning [58]; however, such methods are associated with MPC's computational demand. Critic-only architectures simplify deployment and reduce computational cost [59]; [51]. The applicability of critic-only architecture remains narrower compared to the actor-critic framework. Tackling uncertainty in gain sign of non-affine nonlinear systems, Nussbaum-type critic-only RL has been employed [60]. However, embedded classical control structures, such as PID, into RL frameworks, as seen in Control-Informed RL, have improved adaptability [61]. Actor-Critic methods with dual reward structures have enabled generalization across complex Single Input Single Output (SISO) and Multiple Input and Multiple Output (MIMO) systems [62], and Soft Actor-Critic (SAC) has shown to outperform alternatives in high-dimensional continuous chemical processes though at the expense of high training complexity [63]. In benchmark industrial problems, Neural-Fitted Q-Iteration with Continuous Actions (NFQCA) demonstrated strong disturbance rejection [64]. Additionally, DRL with causal modeling has shown promise in interpretable, energy-efficient supervisory control, yielding significant energy reduction in industrial energy systems [65]. Collectively, these study, including

policy evaluation methods and causal reasoning, continue to expand RL's potential for intelligent, robust, and adaptive control across diverse industrial domains. RL excels in adaptability, while classical controllers still outperform it in terms of guaranteed stability and safety. Hybrid designs, such as Control-Informed RL, attempt to bridge this gap.

The strong coupling between process design and control in chemical systems presents complex optimization challenges, often requiring computationally intensive bi-level Mixed-Integer Non-Linear Programming (MINLP) formulations. Examples include hydrocracking [66] and Electricity-Gas-Heat Integrated Energy Systems (EGH-IES), which exhibit high-dimensional dynamics and multi-energy interactions with variable demands [67], [68]. Traditional methods struggle under such nonlinearities and uncertainties, but RL can help optimize these complex processes. Recent applications span from real-time reactor optimization [5] and chromatography control [69] to Heating, Ventilation, and Air Conditioning (HVAC) system tuning [70] and energy system scheduling [71]. RL has enabled improvements in fuel utilization [72], economic performance [73], yield enhancement [74], and constraint handling [75]. Notably, Pan et al. [76] achieved a 24.9% cost reduction in solvent-switching using PPO, while Oh et al. [77] applied A2C with a surrogate model for adaptive hydrocracking optimization. Integrated Economic MPC-RL frameworks have also been explored for real-time stability and economic efficiency [78], [79]. However, standard Q-learning often underperforms in high-dimensional or continuous domains due to sample inefficiency and safety [80], [81]. To overcome this, advanced methods like DDPG, PPO, and policy gradient variants have been developed to ensure scalable, adaptive, and data-efficient control strategies across process industries [82] – [85]. Also, surrogate-assisted RL can help reduce sample cost but may introduce bias, while transfer/meta-learning enhances generalization across process classes [86][87].

Recent advances have positioned RL as a transformative framework for automating the generation of chemical process flowsheets. Unlike conventional rule-based or heuristic techniques, RL enables sequential decision-making in high-dimensional, constraint-laden environments without requiring explicit domain knowledge. One of the pioneering contributions, SynGameZero by Gotti et al. [88], formulated flowsheet synthesis as a self-play game using Monte Carlo Tree Search (MCTS), achieving high separation efficiency without prior process expertise. Scaling SynGameZero to a more complex flowsheet will make computation difficult. This challenge was addressed by Midgley (2020), who developed a distillation gym. Distillation Gym is an RL environment for designing distillation trains to separate multi-component feed streams, demonstrating RL's applicability to process synthesis, which has traditionally been dominated by optimization techniques like MINLP [89]. Schwaller et al. [90] introduced a transformer-based model combined with a hypergraph exploration strategy for retrosynthetic planning in organic chemistry, enabling richer modeling of multi-step reaction pathways. While their work did not explicitly focus on RL, it provided a foundation for AI-driven synthesis by efficiently representing complex chemical sequences and dependencies. With previous studies not handling recycling streams explicitly, Stops et al., [91], developed a hierarchical RL approach integrated with Graph Neural Networks (GNNs) to handle recycling streams and continuous design variables more effectively. Further innovations by [92] and [93] introduced masked hybrid PPO agents with neural surrogate models, enabling simultaneous design, control, and feasibility assessment under industrial constraints using Aspen Plus integration. Earlier works by Siirola and Rudd [94] laid the foundation with rule-based synthesis frameworks, while recent general-purpose RL agents have demonstrated the ability to discover novel flowsheet topologies beyond predefined heuristics [95], [96]. Efforts to enhance learning efficiency and generalizability include hybrid AI-domain approaches [97], transfer learning frameworks using DWSIM to reuse process knowledge [98], [99] and meta-learning methods designed to adapt RL agents to varying synthesis tasks with improved sample

efficiency [100]. Further study incorporated optimization which defines an optimal flowsheet as a search through the thermodynamic space [101]. RL avoids rigid heuristics of rule-based synthesis but remains sample-inefficient without surrogate integration or transfer learning. A recent extension of modular RL by Gotti et al., [102], further examined the influence of fixed Reflux Ratios (RR), underscoring the importance of flexible action space design in distillation optimization. However, the fixed RR can limit transferability to broader and more adaptive distillation scenarios.

Given the complex, nonlinear, and often turbulent nature of fluid flow, RL has emerged as a compelling alternative to traditional control and design techniques. RL is especially valuable in fluid mechanics, where optimal control and design problems are high-dimensional and analytical solutions are quite impossible to find [103]. Its model-free nature makes RL well-suited for computationally intensive environments, such as fluid dynamics. Recent studies have demonstrated this potential. Kurz et al., [104], integrated RL with high-fidelity Computational Fluid Dynamics (CFD) solvers to enhance turbulence modeling, outperforming classical Smagorinsky models in predictive accuracy and scalability. However, real-time deployment is a significant concern due to its computational complexity. Bae and Koumoutsakos [105] developed a multi-agent RL approach, Scientific Multi-Agent Reinforcement Learning (SciMARL), where agents at grid points learned turbulence behavior near walls, reducing the need for large datasets. In fluid mixing,[106] applied RL to dynamically control stirring in 2D flows, improving mixing uniformity in real-time. Font, [107],s demonstrated RL's effectiveness in suppressing Turbulent Separation Bubbles (TSBs), achieving a 9% reduction in TSB area and a 25.3% drag reduction, surpassing conventional control. Additionally, Ren et al., [108], applied deep RL to active flow control (AFC) around a cylinder, reducing drag by up to 34.2%, which closely approaches the theoretical performance limits. Across these studies, RL has achieved state-of-the-art drag reduction and turbulence suppression. Nonetheless, its high computational cost remains a barrier to practical deployment. Surrogate-assisted or reduced-order models may present promising avenues for accelerating training and real-time deployment.

6. Limitations, Emerging and Future Explorations

Though RL has seen significant success, further research is necessary to bridge the gap between theoretical development and practical industry adoption. Additionally, deploying these solutions in real-world settings and ensuring they generalize effectively across various scenarios remains a pressing challenge in the environment. The majority of works rely heavily on simulation environments, for instance, [76] and [77] do not fully capture the complexities and non-deterministic nature of real industrial systems. The transferability of the RL model from synthetic environments to real-world applications may be affected by model mismatch and noise in plant models. Hence, RL research should be guided towards developing RL algorithms with cross-environment robustness and generalizability to process uncertainty. Solution approaches such as transfer learning and meta-learning are encouraging but require further comprehensive verification for applications across multiple domains [100], [109].

In addition, RL incorporating model developed from first-principles approaches has also been considered. Model-free RL is flexible, but it lacks interpretability. Hybrid models that include physical models or control knowledge in RL frameworks, e.g., in Control-Informed RL, as well as actor-critic models with PID priors, can significantly improve performance and interpretability [61]. An alternative is Surrogate modeling, as in [92], [93], can support the integration of both design and control. Thus, future research should continue along these lines to narrow the gap between domain-knowledge learning.

The use of generative AI especially through text-based transformer model such as large language model (LLM) in flowsheet hasn't been much documented. As presented in section 5, pioneer work by [110] trained a transformer on SFILES 2.0, a textual flowsheet representation, to generate autocompletions of flowsheet topologies. This approach supports interactive design by suggesting plausible unit operations and connections as the user composes a flowsheet.. However, this work is limited to autocompletion and lacks real-world validation, constraint enforcement, or integration with simulation engines. Future research should extend this by fine-tuning domain-specific Large Language Models, LLMs, on broader engineering text and flowsheet corpora; Integrating constraint and feasibility checking, perhaps via hybrid LLM-RL architectures; Coupling text-to-flowsheet generation with simulators (e.g., Aspen Plus), to enable full feedback loops between generation, feasibility evaluation, and optimization [111].

Sample inefficiency and exploration in an unsafe environment are major barriers to deployment in safety-critical systems. Many RL algorithms, especially Q-learning and policy gradient methods, require extensive interactions to understand environment dynamics, which is impractical for real-time systems. Future research should prioritize the development of constraint based RL with safety and effectiveness as priority. Techniques such as reward shaping, Monte Carlo-based policy evaluation, and the use of conservative or constrained policy updates [57], [69] are still underdeveloped.

Finally, the increasing scale and complexity of engineering systems call for attention to multi-agent and distributed reinforcement learning**.** Multi-agent frameworks, such as those employed in turbulence modeling , have shown the ability to handle spatially distributed systems [105]. However, challenges such as coordination, stability, and partial observability remains. There is a need to explore communication strategies and decentralized policies that can work under limited information and real-time constraints.

7. Conclusion

Reinforcement Learning (RL) has demonstrated its versatility and efficacy across diverse engineering disciplines, including process control, optimization, and fluid dynamics. Despite its increasing success, substantial challenges remain, notably in safety, interpretability, and the integration with physical principles. Addressing these issues is a vital direction for future research and development, thereby facilitating the full realization of RL's potential in advancing engineering innovation.

Author contributions: Conceptualization, DOA; investigation, SBG, KKS, AAS; methodology SBG, DOA, and AAS; project administration SBG and DOA, resources, SBG & AAS.; supervision, DOA; visualization, SBG, DOA, AAS; writing original draft preparation, SBG; writing, review and editing, SBG, AAS, KKS, and DOA. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest: The authors declare that there are no conflicts of interest.

8. References

[1] C. Li, P. Zheng, Y. Yin, B. Wang, and L. Wang, "Deep reinforcement learning in smart manufacturing: A review and prospects," *CIRP J. Manuf. Sci. Technol.*, vol. 40, pp. 75–101, 2023, doi: 10.1016/j.cirpj.2022.11.003.

[2] A. G. Barto and R. S. Sutton, "Chapter 19 - Reinforcement Learning in Artificial Intelligence," in *Neural-Network Models of Cognition*, vol. 121, J. W. Donahoe and V. Packard Dorsel, Eds. North-Holland, 1997, pp. 358–386. doi: 10.1016/S0166-4115(97)80105-7.

[3] Z. Yan, F. Xu, J. Tan, H. Liu, and B. Liang, “Reinforcement learning-based integrated active fault diagnosis and tracking control,” *ISA Trans.*, vol. 132, pp. 364–376, Jan. 2023, doi: 10.1016/J.ISATRA.2022.06.020.

[4] N. Rabbani, G. Y. E. Kim, C. J. Suarez, and J. H. Chen, “Applications of machine learning in routine laboratory medicine: Current state and future directions,” *Clin. Biochem.*, vol. 103, pp. 1–7, 2022, doi: 10.1016/j.clinbiochem.2022.02.011.

[5] B. K. M. Powell, D. Machalek, and T. Quah, “Real-time optimization using reinforcement learning,” *Comput. Chem. Eng.*, vol. 143, p. 107077, 2020, doi: 10.1016/j.compchemeng.2020.107077.

[6] D. L. B. Fortela, H. Broussard, R. Ward, C. Broussard, A. P. Mikolajczyk, M. A. Bayoumi, and M. E. Zappi, “Soft Actor-Critic Reinforcement Learning Improves Distillation Column Internals Design Optimization,” *ChemEngineering*, vol. 9, no. 2, pp. 6–10, 2025, doi: 10.3390/chemengineering9020034.

[7] B. Han, Z. Ren, Z. Wu, Y. Zhou, and J. Peng, “Off-Policy Reinforcement Learning with Delayed Rewards,” *Proc. Mach. Learn. Res.*, vol. 162, pp. 8280–8303, Jun. 2021, Accessed: Oct. 02, 2025. [Online]. Available: https://arxiv.org/pdf/2106.11854

[8] R. Nian, J. Liu, and B. Huang, “A review On reinforcement learning : Introduction and applications in industrial process control Artificial intelligence Constrained Markov decision process Dynamic programming,” *Comput. Chem. Eng.*, vol. 139, p. 106886, 2020, doi: 10.1016/j.compchemeng.2020.106886.

[9] E. Muškardin, M. Tappler, B. K. Aichernig, and I. Pill, “Reinforcement Learning under Partial Observability Guided by Learned Environment Models,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 14300 LNCS, pp. 257–276, Jun. 2022, doi: 10.1007/978-3-031-47705-8_14.

[10] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *2010 Int. Conf. Comput. Inf. Syst. Ind. Manag. Appl. CISIM 2010*, vol. 292, pp. 228–232, 1992, doi: 10.1109/CISIM.2010.5643660.

[11] D. P. Bertsekas and J. N. Tsitsiklis, “Neuro-dynamic programming: an overview,” *Proc. 1995 34th IEEE Conf. Decis. Control*, vol. 1, pp. 560–564, 1995, doi: 10.1109/CDC.1995.478953.

[12] J. F. Cevallos M., A. Rizzardi, S. Sicari, and A. Coen Porisini, “Deep Reinforcement Learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges,” *Comput. Networks*, vol. 236, p. 110016, 2023, doi: 10.1016/j.comnet.2023.110016.

[13] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” pp. 1–12, 2017, doi: 10.48550/arXiv.1707.06347.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, Sep. 2015, Accessed: Oct. 06, 2025. [Online]. Available: https://arxiv.org/pdf/1509.02971

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 5, pp. 2976–2989, Jan. 2018, Accessed: Oct. 06, 2025. [Online]. Available:

https://arxiv.org/pdf/1801.01290

[16] S. Gros and M. Zanon, "Data-driven Economic NMPC using Reinforcement Learning," *IEEE Trans. Automat. Contr.*, vol. 65, no. 2, pp. 636–648, Apr. 2019, doi: 10.1109/tac.2019.2913768.

[17] O. Dogru, J. Xie, O. Prakash, R. Chiplunkar, J. Soesanto, H. Chen, K. Velswamy, F. Ibrahim, B. Huang, O. Dogru, J. Xie, O. Prakash, R. Chiplunkar, J. Soesanto, H. Chen, K. Velswamy, F. Ibrahim, and B. Huang, "Reinforcement Learning in Process Industries: Review and Perspective," *IEEE/CAA J. Autom. Sin. 2024, Vol. 11, Issue 2, Pages 283-300*, vol. 11, no. 2, pp. 283–300, Feb. 2024, doi: 10.1109/JAS.2024.124227.

[18] D. Weinberg, Q. Wang, T. O. Timoudas, and C. Fischione, "A Review of Reinforcement Learning for Controlling Building Energy Systems From a Computer Science Perspective," *Sustain. Cities Soc.*, vol. 89, p. 104351, Feb. 2023, doi: 10.1016/J.SCS.2022.104351.

[19] R. S. . & B. A. G. Sutton, "Reinforcement Learning, second edition: An Introduction - Richard S. Sutton, Andrew G. Barto - Google Books," *An introduction. MIT press.*, 2018. https://books.google.ca/books?hl=en&lr=&id=sWV0DwAAQBAJ&oi=fnd&pg=PR7&ots=1-9fp3csTe&sig=ETNGk3S1akuGK8jwHHxkHBcLC-w#v=onepage&q&f=false (accessed Feb. 10, 2025).

[20] D. Wang and R. Snooks, "Artificial Intuitions of Generative Design: An Approach Based on Reinforcement Learning," *Proc. 2020 Digit.*, pp. 189–198, 2021, doi: 10.1007/978-981-33-4400-6_18.

[21] T. Westenbroekf, A. Agrawalf, F. Castaneda, S. S. Sastry, and K. Sreenath, "Combining Model-Based Design and Model-Free Policy Optimization to Learn Safe, Stabilizing Controllers," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 19–24, Jan. 2021, doi: 10.1016/J.IFACOL.2021.08.468.

[22] L. Zou, "Meta-reinforcement learning," *Meta-Learning*, pp. 267–297, 2023, doi: 10.1016/B978-0-323-89931-4.00011-0.

[23] M. C. Mckenzie and M. D. Mcdonnell, "Modern Value Based Reinforcement Learning: A Chronological Review," *IEEE Access*, vol. 10, pp. 134704–134725, 2022, doi: 10.1109/ACCESS.2022.3228647.

[24] A. Plaat, "Policy-Based Reinforcement Learning," *Deep Reinf. Learn.*, pp. 101–133, 2022, doi: 10.1007/978-981-19-0638-1_4.

[25] J. Zhu, H. Zhang, and Z. Pan, "Value-Based Continuous Control Without Concrete State-Action Value Function.," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12690 LNCS, pp. 352–364, Jul. 2021, doi: 10.1007/978-3-030-78811-7_34.

[26] F. Scharf, F. Helfenstein, and J. Jäger, "Actor vs Critic: Learning the Policy or Learning the Value," *Stud. Comput. Intell.*, vol. 883, pp. 123–133, Jan. 2021, doi: 10.1007/978-3-030-41188-6_11.

[27] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," *30th AAAI Conf. Artif. Intell. AAAI 2016*, pp. 1934–1940, 2016, doi: 10.1609/AAAI.V30I1.10226.

[28] S. Brandi, M. Fiorentini, and A. Capozzoli, “Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management,” *Autom. Constr.*, vol. 135, p. 104128, Mar. 2022, doi: 10.1016/J.AUTCON.2022.104128.

[29] O. Dogru, N. Wieczorek, K. Velswamy, F. Ibrahim, and B. Huang, “Online reinforcement learning for a continuous space system with experimental validation,” *J. Process Control*, vol. 104, pp. 86–100, Aug. 2021, doi: 10.1016/J.JPROCONT.2021.06.004.

[30] X. Lou, Q. Yin, J. Zhang, C. Yu, Z. He, N. Cheng, and K. Huang, “Offline reinforcement learning with representations for actions,” *Inf. Sci. (Ny).*, vol. 610, pp. 746–758, Sep. 2022, doi: 10.1016/J.INS.2022.08.019.

[31] E. A. G. de Souza, M. S. Nagano, and G. A. Rolim, “Dynamic Programming algorithms and their applications in machine scheduling: A review,” *Expert Syst. Appl.*, vol. 190, p. 116180, Mar. 2022, doi: 10.1016/J.ESWA.2021.116180.

[32] Q. Wei, Z. Yang, H. Su, and L. Wang, “Monte Carlo-based reinforcement learning control for unmanned aerial vehicle systems,” *Neurocomputing*, vol. 507, pp. 282–291, Oct. 2022, doi: 10.1016/J.NEUCOM.2022.08.011.

[33] Y. Ma, J. Feng, Q. Wu, R. Zheng, J. Zhu, J. Xi, and M. Zhang, “Provable causal distributed two-time-scale temporal-difference learning with instrumental variables,” *Expert Syst. Appl.*, vol. 287, p. 128187, Aug. 2025, doi: 10.1016/J.ESWA.2025.128187.

[34] M. K. Gupta, N. Hemachandra, and S. Bhatnagar, “Learning in sequential decision-making under uncertainty,” *Artif. Intell. Mach. Learn. EDGE Comput.*, pp. 75–85, Jan. 2022, doi: 10.1016/B978-0-12-824054-0.00011-3.

[35] M. Sewak, “Temporal Difference Learning, SARSA, and Q-Learning,” in *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*, Singapore: Springer Singapore, 2019, pp. 51–63. doi: 10.1007/978-981-13-8285-7_4.

[36] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, “Q-Learning Algorithms: A Comprehensive Classification and Applications,” *IEEE Access*, vol. 7, pp. 133653–133667, 2019, doi: 10.1109/ACCESS.2019.2941229.

[37] Y. Huang, X. Wan, L. Zhang, and X. Lu, “A novel deep reinforcement learning framework with BiLSTM-Attention networks for algorithmic trading,” *Expert Syst. Appl.*, vol. 240, p. 122581, Apr. 2024, doi: 10.1016/J.ESWA.2023.122581.

[38] B. Hengst, “Hierarchical Reinforcement Learning,” *Encycl. Mach. Learn.*, pp. 495–502, 2011, doi: 10.1007/978-0-387-30164-8_363.

[39] S. Pateria, B. Subagdja, A. H. Tan, and C. Quek, “Hierarchical Reinforcement Learning: A Comprehensive Survey,” *ACM Comput. Surv.*, vol. 54, no. 5, Jun. 2021, doi: 10.1145/3453160.

[40] B. Xu, Q. Zhou, J. Shi, and S. Li, “Hierarchical Q-learning network for online simultaneous optimization of energy efficiency and battery life of the battery/ultracapacitor electric vehicle,” *J. Energy Storage*, vol. 46, p. 103925, Feb. 2022, doi: 10.1016/J.EST.2021.103925.

[41] J. Hu and M. P. Wellman, “Nash Q-learning for general-sum stochastic games,” *J. Mach. Learn. Res.*, vol. 4, no. 6, pp. 1039–1069, Aug. 2004, doi: 10.1162/1532443041827880.

[42] X. Zhang, D. Meng, J. Cai, G. Zhang, T. Yu, F. Pan, and Y. Yang, "A swarm based double Q-learning for optimal PV array reconfiguration with a coordinated control of hydrogen energy storage system," *Energy*, vol. 266, p. 126483, Mar. 2023, doi: 10.1016/J.ENERGY.2022.126483.

[43] M. Lehmann, "The Definitive Guide to Policy Gradients in Deep Reinforcement Learning: Theory, Algorithms and Implementations," 2024, [Online]. Available: http://arxiv.org/abs/2401.13662

[44] M. Paczkowski, "Low-friction composite creping blades improve tissue properties," *Pulp Pap.*, vol. 70, no. 9, 1996.

[45] J. Zhang, J. Kim, B. O'Donoghue, and S. Boyd, "Sample Efficient Reinforcement Learning with REINFORCE," *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 12B, pp. 10887–10895, 2021, doi: 10.1609/aaai.v35i12.17300.

[46] E. H. Sumiea, S. J. Abdulkadir, H. S. Alhussian, S. M. Al-Selwi, A. Alqushaibi, M. G. Ragab, and S. M. Fati, "Deep deterministic policy gradient algorithm: A systematic review," *Heliyon*, vol. 10, no. 9, p. e30697, 2024, doi: 10.1016/j.heliyon.2024.e30697.

[47] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *31st Int. Conf. Mach. Learn. ICML 2014*, vol. 1, pp. 605–619, 2014.

[48] OpenAI, "Deep Deterministic Policy Gradient — Spinning Up documentation." https://spinningup.openai.com/en/latest/algorithms/ddpg.html (accessed May 30, 2025).

[49] H. A. Neamah and O. A. Mayorga, "Optimized TD3 Algorithm for Robust Autonomous Navigation in Crowded and Dynamic Human-Interaction Environments," *Results Eng.*, vol. 24, p. 102874, Dec. 2024, doi: 10.1016/j.rineng.2024.102874.

[50] U. Saha, A. Jawad, S. Shahria, and A. B. M. H. U. Rashid, "Proximal policy optimization-based reinforcement learning approach for DC-DC boost converter control: A comparative evaluation against traditional control techniques," *Heliyon*, vol. 10, no. 18, p. e37823, Sep. 2024, doi: 10.1016/J.HELIYON.2024.E37823.

[51] L. Xue, Z. G. Liu, and W. Zhang, "Adaptive stabilization of stochastic systems with polynomial nonlinear conditions, unknown parameters, and dead-zone actuator," *Int. J. Robust Nonlinear Control*, vol. 34, no. 8, pp. 5524–5539, May 2024, doi: 10.1002/RNC.7254;WGROUP:STRING:PUBLICATION.

[52] K. Ammari and G. Bel Mufti, "Controlling a Dynamic System Through Reinforcement Learning," *Trends Math.*, vol. Part F1467, pp. 23–30, 2023, doi: 10.1007/978-3-031-35675-9_2.

[53] Z. Deng, X. Huo, Q. Du, and Q. Liu, "Reinforcement Learning-based Data-driven Control Design for Motion Control Systems," *Proc. 36th Chinese Control Decis. Conf. CCDC 2024*, pp. 5745–5749, 2024, doi: 10.1109/CCDC62350.2024.10587783.

[54] T. Binazadeh and M. Ali Rahgoshay, "Robust output tracking of a class of non-affine systems," *Syst. Sci. Control Eng.*, vol. 5, no. 1, pp. 426–433, Jan. 2017, doi: 10.1080/21642583.2017.1376296.

[55] Z. Shao, Y. Wang, J. Lai, and S. Wang, "Robust tracking control of non-affine systems under

unknown control directions and uncertain trajectory," *Proc. - 2020 Chinese Autom. Congr. CAC 2020*, pp. 7089–7094, Nov. 2020, doi: 10.1109/CAC51589.2020.9326902.

[56] M. Szuster and Z. Hendzel, "Reinforcement Learning in the Control of Nonlinear Continuous Systems," in *Intelligent Optimal Adaptive Control for Mechatronic Systems*, Cham: Springer International Publishing, 2018, pp. 255–297. doi: 10.1007/978-3-319-68826-8_8.

[57] H. Yoo, B. Kim, J. W. Kim, and J. H. Lee, "Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation," *Comput. Chem. Eng.*, vol. 144, p. 107133, 2021, doi: 10.1016/j.compchemeng.2020.107133.

[58] Q. Han, F. Boussaid, and M. Bennamoun, "Model Predictive Control-Based Reinforcement Learning," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2024, pp. 1–5. doi: 10.1109/ISCAS58744.2024.10558623.

[59] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-Learning-Based Robust Controller Design for Continuous-Time Uncertain Nonlinear Systems Subject to Input Constraints," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, 2015, doi: 10.1109/TCYB.2015.2417170.

[60] X. Song, M. Huang, G. Wen, L. Ma, J. Yao, and Z. Lu, "Reinforcement learning-based control for a class of nonlinear systems with unknown control directions," *Chinese Control Conf. CCC*, vol. 2019-July, pp. 2519–2524, 2019, doi: 10.23919/ChiCC.2019.8865292.

[61] M. Bloor, A. Ahmed, N. Kotecha, M. Mercangöz, C. Tsay, and E. A. D. R. Chanona, "Control-Informed Reinforcement Learning for Chemical Processes," 2025, doi: 10.1021/acs.iecr.4c03233.

[62] S. Spielberg, P. Kumar, and B. Gopaluni, "Process Control using Deep Reinforcement Learning".

[63] M. Wu, F. Elmaz, U. Di Caprio, D. De Clercq, S. Mercelis, P. Hellinckx, L. Braeken, F. Vermeire, and M. E. Leblebici, "Real-time optimization of a chemical plant with continuous flow reactors via reinforcement learning," *Comput. Aided Chem. Eng.*, vol. 52, pp. 457–462, Jan. 2023, doi: 10.1016/B978-0-443-15274-0.50073-1.

[64] R. Hafner and M. Riedmiller, "Challenges and benchmarks from technical process control," pp. 137–169, 2011, doi: 10.1007/s10994-011-5235-x.

[65] K. Nadim, M.-S. Ouali, H. Ghezzaz, and A. Ragab, "Learn-to-supervise: Causal reinforcement learning for high-level control in industrial processes," *Eng. Appl. Artif. Intell.*, vol. 126, p. 106853, 2023, doi: 10.1016/j.engappai.2023.106853.

[66] W. Song, V. Mahalec, J. Long, M. Yang, and F. Qian, "Modeling the Hydrocracking Process with Deep Neural Networks," *Ind. Eng. Chem. Res.*, vol. 59, no. 7, pp. 3077–3090, Feb. 2020, doi: 10.1021/acs.iecr.9b06295.

[67] C. Qin, Q. Yan, and G. He, "Integrated energy systems planning with electricity, heat and gas using particle swarm optimization," *Energy*, vol. 188, p. 116044, Dec. 2019, doi: 10.1016/J.ENERGY.2019.116044.

[68] H. Faramarzi, N. Ghaffarzadeh, and F. Shahnia, "Intelligent Management of Integrated Energy

Systems with a Stochastic Multi-Objective Approach with Emphasis on Demand Response, Energy Storage Devices, and Power-to-Gas," *Sustain. 2025, Vol. 17, Page 3001*, vol. 17, no. 7, p. 3001, Mar. 2025, doi: 10.3390/SU17073001.

[69] S. Nikita, A. Tiwari, D. Sonawat, H. Kodamana, and A. S. Rathore, "Reinforcement learning based optimization of process chromatography for continuous processing of biopharmaceuticals," *Chem. Eng. Sci.*, vol. 230, p. 116171, 2021, doi: 10.1016/j.ces.2020.116171.

[70] E. Ortiz-Mansilla, J. J. García-Esteban, J. Bravo-Abad, and J. C. Cuevas, "Deep reinforcement learning for radiative heat transfer optimization problems," *Phys. Rev. Appl.*, vol. 22, no. 5, Aug. 2024, doi: 10.1103/PhysRevApplied.22.054071.

[71] F. Li, L. Liu, and Y. Yu, "Deep reinforcement learning-based multi-objective optimization for electricity–gas–heat integrated energy systems," *Expert Syst. Appl.*, vol. 262, p. 125558, Mar. 2025, doi: 10.1016/J.ESWA.2024.125558.

[72] Y. Cheng, Y. Huang, B. Pang, and W. Zhang, "ThermalNet: A deep reinforcement learning-based combustion optimization system for coal-fired boiler," *Eng. Appl. Artif. Intell.*, vol. 74, pp. 303–311, 2018.

[73] T. Quah, D. Machalek, and K. M. Powell, "Comparing reinforcement learning methods for real-time optimization of a chemical process," *Processes*, vol. 8, no. 11, pp. 1–19, 2020, doi: 10.3390/pr8111497.

[74] K. Alhazmi, F. Albalawi, and S. M. Sarathy, "A reinforcement learning-based economic model predictive control framework for autonomous operation of chemical reactors," *Chem. Eng. J.*, vol. 428, no. July 2021, p. 130993, 2022, doi: 10.1016/j.cej.2021.130993.

[75] E. Pan, P. Petsagkourakis, M. Mowbray, D. Zhang, and A. del Rio-Chanona, "Constrained Q-learning for batch process optimization," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 492–497, 2021, doi: 10.1016/j.ifacol.2021.08.290.

[76] F. Elmaz, U. Di Caprio, M. Wu, Y. Wouters, G. Van Der Vorst, N. Vandervoort, A. Anwar, M. E. Leblebici, P. Hellinckx, and S. Mercelis, "Reinforcement learning-based approach for optimizing solvent-switch processes," *Comput. Chem. Eng.*, vol. 176, p. 108310, 2023, doi: 10.1016/j.compchemeng.2023.108310.

[77] D. H. Oh, D. Adams, N. D. Vo, D. Q. Gbadago, C. H. Lee, and M. Oh, "Actor-critic reinforcement learning to estimate the optimal operating conditions of the hydrocracking process," *Comput. Chem. Eng.*, vol. 149, p. 107280, 2021, doi: 10.1016/j.compchemeng.2021.107280.

[78] Z. Zhang and S. Li, "Enhanced reinforcement learning in two-layer economic model predictive control for operation optimization in dynamic environment," *Chem. Eng. Res. Des.*, vol. 196, pp. 133–143, Aug. 2023, doi: 10.1016/J.CHERD.2023.06.023.

[79] S. Sachio, A. E. del-Rio Chanona, and P. Petsagkourakis, "Simultaneous Process Design and Control Optimization using Reinforcement Learning," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 510–515, 2021, doi: 10.1016/j.ifacol.2021.08.293.

[80] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen, "Sample Complexity of Asynchronous Q-Learning: Sharper Analysis and Variance Reduction," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 448–

473, Jan. 2022, doi: 10.1109/TIT.2021.3120096.

[81] M. Neumann and D. S. Palkovits, “Reinforcement Learning Approaches for the Optimization of the Partial Oxidation Reaction of Methane,” *Ind. Eng. Chem. Res.*, vol. 61, no. 11, pp. 3910–3916, Mar. 2022, doi: 10.1021/ACS.IECR.1C04622.

[82] J. Arshad, A. Khan, M. Aftab, M. Hussain, A. U. Rehman, S. Ahmad, A. M. Al-Shayea, and M. Shafiq, “Deep Deterministic Policy Gradient to Regulate Feedback Control Systems Using Reinforcement Learning,” *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1153–1169, Oct. 2021, doi: 10.32604/CMC.2022.021917.

[83] K. Moghaddasi, S. Rajabi, F. Soleimanian Gharehchopogh, and M. Hosseinzadeh, “An Energy-Efficient Data Offloading Strategy for 5G-Enabled Vehicular Edge Computing Networks Using Double Deep Q-Network,” *Wirel. Pers. Commun.*, vol. 133, no. 3, pp. 2019–2064, Dec. 2023, doi: 10.1007/S11277-024-10862-5/METRICS.

[84] F. Tavakkoli, P. Sarhadi, B. Clement, and W. Naeem, “Model Free Deep Deterministic Policy Gradient Controller for Setpoint Tracking of Non-Minimum Phase Systems,” *2024 UKACC 14th Int. Conf. Control. Control 2024*, pp. 163–168, 2024, doi: 10.1109/CONTROL60310.2024.10531953.

[85] L. Fernando, B. L. P. Lik, C. Yuen, and U. X. Tan, “A Scalable Decentralized Reinforcement Learning Framework for UAV Target Localization Using Recurrent PPO,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 129–132, 2024, doi: 10.1109/TENCON61640.2024.10902783.

[86] O. Francon, S. Gonzalez, B. Hodjat, E. Meyerson, R. Miikkulainen, X. Qiu, and H. Shahrzad, “Effective Reinforcement Learning through Evolutionary Surrogate-Assisted Prescription,” *GECCO 2020 - Proc. 2020 Genet. Evol. Comput. Conf.*, pp. 814–822, Feb. 2020, doi: 10.1145/3377930.3389842.

[87] D. G. McClement, N. P. Lawrence, P. D. Loewen, M. G. Forbes, J. U. Backström, and R. B. Gopaluni, “A meta-reinforcement learning approach to process control,” *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 685–692, 2021, doi: 10.1016/j.ifacol.2021.08.321.

[88] Q. Göttl, D. G. Grimm, and J. Burger, “Automated synthesis of steady-state continuous processes using reinforcement learning,” *Front. Chem. Sci. Eng.*, vol. 16, no. 2, pp. 288–302, 2022, doi: 10.1007/s11705-021-2055-9.

[89] L. I. Midgley, “Deep Reinforcement Learning for Process Synthesis,” *CoRR*, vol. abs/2009.1, 2020, [Online]. Available: https://arxiv.org/abs/2009.13265

[90] P. Schwaller, R. Petraglia, V. Zullo, V. H. Nair, R. A. Haeuselmann, R. Pisoni, C. Bekas, A. Iuliano, and T. Laino, “Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy,” *pubs.rsc.orgP Schwaller, R Petraglia, V Zullo, VH Nair, RA Haeuselmann, R Pisoni, C Bekas, A IulianoChemical Sci. 2020•pubs.rsc.org*, 2020, doi: 10.1039/c9sc05704h.

[91] L. Stops, R. Leenhouts, Q. Gao, and A. M. Schweidtmann, “Flowsheet generation through hierarchical reinforcement learning and graph neural networks,” *AIChE J.*, vol. 69, no. 1, pp. 1–22, 2023, doi: 10.1002/aic.17938.

[92] S. Reynoso-Donzelli and L. A. Ricardez-Sandoval, "A reinforcement learning approach with masked agents for chemical process flowsheet design," *AIChE J.*, no. August 2024, pp. 1–16, 2024, doi: 10.1002/aic.18584.

[93] S. Reynoso-Donzelli and L. A. Ricardez-Sandoval, "An integrated reinforcement learning framework for simultaneous generation, design, and control of chemical process flowsheets," *Comput. Chem. Eng.*, vol. 194, no. December 2024, p. 108988, 2025, doi: 10.1016/j.compchemeng.2024.108988.

[94] J. J. Siirola and D. F. Rudd, "Computer-Aided Synthesis of Chemical Process Designs From Reaction Path Data to the Process Task Network," *Ind. Eng. Chem. Fundam.*, vol. 10, no. 3, pp. 353–362, 1971, doi: 10.1021/i160039a003.

[95] S. C. P. A. van Kalmthout, L. I. Midgley, and M. B. Franke, "Synthesis of separation processes with reinforcement learning," *arXiv*, pp. 1–18, 2022, [Online]. Available: http://arxiv.org/abs/2211.04327

[96] N. K. Brown, A. P. Garland, G. M. Fadel, and G. Li, "Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains," *Mater. Des.*, vol. 218, p. 110672, 2022, doi: 10.1016/j.matdes.2022.110672.

[97] V. Mann, M. Sales-Cruz, R. Gani, and V. Venkatasubramanian, "eSFILES: Intelligent process flowsheet synthesis using process knowledge, symbolic AI, and machine learning," *Comput. Chem. Eng.*, vol. 181, pp. 1–41, 2024, doi: 10.1016/j.compchemeng.2023.108505.

[98] A. Farooq and K. Iqbal, "A Survey of Reinforcement Learning for Optimization in Automation," *IEEE Int. Conf. Autom. Sci. Eng.*, pp. 2487–2494, Aug. 2024, doi: 10.1109/CASE59546.2024.10711718.

[99] Q. Gao, H. Yang, S. M. Shanbhag, and A. M. Schweidtmann, "Transfer learning for process design with reinforcement learning," *Comput. Aided Chem. Eng.*, vol. 52, pp. 2005–2010, Feb. 2023, doi: 10.1016/B978-0-443-15274-0.50319-X.

[100] Q. Gao, H. Yang, M. F. Theisen, and A. M. Schweidtmann, "Accelerating process synthesis with reinforcement learning: Transfer learning from multi-fidelity simulations and variational autoencoders," *Comput. Chem. Eng.*, p. 109192, May 2025, doi: 10.1016/J.COMPCHEMENG.2025.109192.

[101] A. Khan and A. Lapkin, "Searching for optimal process routes: A reinforcement learning approach," *Comput. Chem. Eng.*, vol. 141, p. 107027, Oct. 2020, doi: 10.1016/J.COMPCHEMENG.2020.107027.

[102] Q. Göttl, J. Pirnay, J. Burger, and D. G. Grimm, "Deep reinforcement learning enables conceptual design of processes for separating azeotropic mixtures without prior knowledge," *Comput. Chem. Eng.*, vol. 194, p. 108975, Mar. 2025, doi: 10.1016/J.COMPCHEMENG.2024.108975.

[103] Y. Wang, P. K. Jimack, M. A. Walkley, D. Yang, and H. M. Thompson, "An optimal control method for time-dependent fluid-structure interaction problems," *Struct. Multidiscip. Optim.*, vol. 64, no. 4, pp. 1939–1962, Oct. 2021, doi: 10.1007/S00158-021-02956-6/FIGURES/32.

[104] M. Kurz, P. Offenhäuser, D. Viola, O. Shcherbakov, M. Resch, and A. Beck, "Deep

reinforcement learning for computational fluid dynamics on HPC systems," *J. Comput. Sci.*, vol. 65, no. October, p. 101884, 2022, doi: 10.1016/j.jocs.2022.101884.

[105] H. J. Bae and P. Koumoutsakos, "Scientific multi-agent reinforcement learning for wall-models of turbulent flows," *Nat. Commun.*, vol. 13, no. 1, pp. 1–9, 2022, doi: 10.1038/s41467-022-28957-7.

[106] M. Konishi, M. Inubushi, and S. Goto, "Fluid mixing optimization with reinforcement learning," *Sci. Rep.*, vol. 12, no. 1, pp. 1–8, 2022, doi: 10.1038/s41598-022-18037-7.

[107] B. Font, "Deep reinforcement learning for active flow control in a turbulent separation bubble," *Nat. Commun.*, no. June 2024, 2025, doi: 10.1038/s41467-025-56408-6.

[108] F. Ren, J. Rabault, and H. Tang, "Applying deep reinforcement learning to active flow control in weakly turbulent conditions," *Phys. Fluids*, vol. 33, no. 3, 2021, doi: 10.1063/5.0037371.

[109] J. Gao, A. Wahlen, C. Ju, Y. Chen, G. Lan, and Z. Tong, "Reinforcement learning-based control for waste biorefining processes under uncertainty," *Commun. Eng.*, vol. 3, no. 1, pp. 1–10, 2024, doi: 10.1038/s44172-024-00183-7.

[110] G. Vogel, L. Schulze Balhorn, and A. M. Schweidtmann, "Learning from flowsheets: A generative transformer model for autocompletion of flowsheets," *Comput. Chem. Eng.*, vol. 171, Mar. 2023, doi: 10.1016/j.compchemeng.2023.108162.

[111] G. Vogel, L. Schulze Balhorn, and A. M. Schweidtmann, "Learning from flowsheets: A generative transformer model for autocompletion of flowsheets," *Comput. Chem. Eng.*, vol. 171, p. 108162, Mar. 2023, doi: 10.1016/J.COMPCHEMENG.2023.108162.