On the Limitations of Safe Reinforcement Learning for Unstable Chemical Process Control

José Rodrigues Torraca Neto[a,*], Bruno Didier Olivier Capron[a], Argimiro Resende Secchi[b], Ehecatl Antonio del Rio Chanona[c]

[a] *Universidade Federal do Rio de Janeiro, Escola de Química, Rio de Janeiro, RJ, Brasil*
[b] *Universidade Federal do Rio de Janeiro, Programa de Engenharia Química, Rio de Janeiro, RJ, Brasil*
[c] *Imperial College London, Department of Chemical Engineering, London, United Kingdom*
[*] *joseneto@eq.ufrj.br*

ABSTRACT

Safe reinforcement learning (Safe RL) has emerged to address key limitations of standard reinforcement learning (RL), particularly its lack of safety guarantees during exploration and execution, which restricts its use in safety-critical systems such as chemical processes. In this work, we evaluate state-of-the-art Safe RL algorithms on an unstable continuous stirred tank reactor (CSTR) benchmark. All safe methods significantly reduce violations compared to unconstrained PPO, confirming the effectiveness of constraint-aware formulations. However, a clear trade-off between performance and safety persists: methods that prioritize reward tend to exhibit weaker constraint reliability, while those that emphasize safety achieve more consistent constraint satisfaction at the expense of performance. This behavior reflects the conflict between competing objectives in the constrained Markov decision process (CMDP) framework. Among the evaluated methods, PPO-Lag achieves the best overall performance, with the highest reward, lowest cost, and fewest constraint violations, while PCPO provides a balanced and stable safety–performance trade-off. These results highlight the limitations of current Safe RL approaches and motivate the development of more reliable control methods.

Keywords: Safe reinforcement learning; Constrained Markov decision processes; Process safety

1 Introduction

Reinforcement learning (RL) has gained attention in chemical process control due to its ability to handle nonlinear, high-dimensional, and partially unknown systems. Unlike model-based approaches such as nonlinear model predictive control (NMPC), RL enables direct policy optimization through interaction, often in simulated environments. However, its application in safety-critical settings remains challenging, as constraint satisfaction must be ensured during both learning and execution. Safe reinforcement learning (Safe RL) addresses this by incorporating safety constraints into the learning process. Recent surveys (Brunke et al., 2022; Gu et al., 2024) classify methods into constrained optimization, shielding, Lyapunov-based, and risk-sensitive approaches. A common formulation is the constrained Markov decision process (CMDP), where policies are optimized under expected cost constraints. Practical methods rely on Lagrangian relaxation, primal–dual updates, or trust-region schemes, as implemented in frameworks such as OmniSafe (Ji et al., 2024).

Despite these advances, applying Safe RL to unstable systems remains difficult. Chemical processes are inherently nonlinear, often exhibiting input multiplicity, strong coupling, and open-loop instability, which lead to narrow feasible operating regions bounded by unsafe regimes. Exothermic CSTRs are a representative example of these challenges, in which thermal runaway and constraint violations can arise from small perturbations. From a control perspective, stability guarantees typically rely on Lyapunov-based designs or persistent excitation (Annaswamy, 2023). In contrast, standard RL optimizes expected return without explicit closed-loop stability guarantees. This mismatch is critical in unstable systems, where small deviations can lead to unsafe behavior. Reward functions that favor high performance may drive the system toward instability, leading to a fundamental trade-off between performance and safety.

In this work, we evaluate representative Safe RL algorithms, including PPO-Lag (Ray et al., 2019), PCPO (Yang et al., 2020), FOCOPS (Y. Zhang et al., 2020), and P3O (L. Zhang et al., 2022), on an unstable CSTR benchmark, focusing on enforcing safety constraints while maintaining control performance under nonlinear dynamics and disturbances.

2 Methodology

2.1 *Problem Formulation as a CMDP*

Reinforcement learning (RL) traditionally seeks to maximize the expected cumulative reward:

$$\max_{\pi_\theta} J_r(\pi_\theta) \tag{1}$$

where $\pi_\theta(a|s)$ is a parameterized policy mapping states $s \in \mathscr{S}$ to actions $a \in \mathscr{A}$, and $J_r(\pi_\theta)$ denotes the expected discounted return under policy π_θ, which encodes the control objective (e.g., tracking performance or economic profit). In safety-critical settings, this objective is augmented with a cost signal that captures constraint violations or risk. This leads to the Constrained Markov Decision Process (CMDP) formulation, defined by $(\mathscr{S}, \mathscr{A}, P, r, c)$, where $\mathscr{S}$ and $\mathscr{A}$ are the state and action spaces, $P(s'|s,a)$ is the transition probability, and $r(s,a)$ and $c(s,a)$ are the reward and cost functions, respectively. The goal is to maximize performance while limiting unsafe behavior:

$$\max_{\pi_\theta} J_r(\pi_\theta) \quad \text{s.t.} \quad J_c(\pi_\theta) \leq d \tag{2}$$

where J_r and J_c denote the expected discounted reward and cost, respectively, and d is a predefined cost limit. This formulation introduces a fundamental trade-off: reward drives performance improvement, while cost penalizes unsafe operation. In the CSTR setting, r encodes control objectives (e.g., setpoint tracking), whereas c penalizes unsafe operating conditions, such as excessive temperature that can lead to thermal runaway.

2.2 *Safe Reinforcement Learning Algorithms*

We evaluate on-policy Safe RL methods that differ in how they enforce CMDP constraints, including Lagrangian, trust-region, and penalty-based approaches (Table 1).

Table 1: Classification of evaluated Safe RL algorithms

Domain	Type	Algorithms
On-Policy	Primal-Dual (Lagrangian)	PPO-Lag
On-Policy	Convex Optimization	PCPO, FOCOPS
On-Policy	Penalty Function	P3O

2.3 *Lagrangian Methods: PPO-Lag (Ray et al., 2019)*

Lagrangian methods convert the constrained problem into an unconstrained one using a dual variable λ:

$$\mathscr{L}(\theta, \lambda) = J_r(\pi_\theta) - \lambda(J_c(\pi_\theta) - d) \tag{3}$$

Policy optimization and constraint satisfaction are handled via a saddle-point problem, with λ updated as:

$$\lambda \leftarrow [\lambda + \eta(J_c(\pi_\theta) - d)]^+ \tag{4}$$

In PPO-Lag, this corresponds to replacing the advantage with a penalized form:

$$\hat{A}_t = \hat{A}_t^r - \lambda \hat{A}_t^c \tag{5}$$

where $\hat{A}_t^r$ and $\hat{A}_t^c$ are estimates of the reward and cost advantages at time step t, respectively, representing the relative benefit of an action with respect to performance and constraint violation.

2.4 *Convex Optimization Methods: PCPO (Yang et al., 2020), FOCOPS (Y. Zhang et al., 2020)*

PCPO separates reward improvement and constraint enforcement. First, a trust-region step improves reward; if constraints are violated, the update is projected back onto the feasible set:

$$\min_{\Delta\theta} \tfrac{1}{2}\Delta\theta^T H \Delta\theta \quad \text{s.t.} \quad c^T\Delta\theta + b \leq 0 \tag{6}$$

where H is an approximation of the Hessian matrix of the objective, arising from second-order optimization methods, c denotes the gradient of the constraint function with respect to the policy parameters, and b represents the

current level of constraint violation. **FOCOPS** avoids second-order methods by optimizing directly in policy space. The optimal update is:

$$\pi^*(a|s) \propto \pi_k(a|s) \exp\left(\tfrac{1}{\eta}(\hat{A}^r - \lambda \hat{A}^c)\right) \tag{7}$$

and is projected back to the parameterized policy via Kullback–Leibler (KL) divergence minimization. The resulting gradient update is:

$$\nabla_\theta L(\theta) = \mathbb{E}_t \left[\nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{A}_t^r - \lambda \hat{A}_t^c\right)\right] \tag{8}$$

2.5 *Penalty-Based Methods: P3O (L. Zhang et al., 2022)*

Penalty-based methods incorporate constraint violations directly into the objective:

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t,a_t) - \beta c(s_t,a_t)\right)\right] \tag{9}$$

This leads to a penalized advantage:

$$\hat{A}_t = \hat{A}_t^r - \beta \hat{A}_t^c \tag{10}$$

Unlike Lagrangian methods, the penalty coefficient β is fixed, making performance sensitive to tuning.

2.6 *Implementation*

All algorithms are implemented using the OmniSafe framework (Ji et al., 2024), ensuring consistent benchmarking. The unstable CSTR model is implemented using the Gymnasium framework (Towers et al., 2024).

2.6.1 *Process Model*

The system is a continuous stirred tank reactor with state: $x = [C_A,\ T]^T$. The dynamics are:

$$\frac{dC_A}{dt} = \frac{q}{V}(C_{A,f} - C_A) - k(T)C_A, \quad \frac{dT}{dt} = \frac{q}{V}(T_f - T) + \frac{-\Delta H}{\rho C_p}k(T)C_A + \frac{UA}{V\rho C_p}(T_c - T) \tag{11}$$

with Arrhenius kinetics:

$$k(T) = k_0 \exp\left(-\frac{E}{RT}\right) \tag{12}$$

The system is integrated using a fourth-order Runge–Kutta scheme. The control problem is regulatory, with fixed setpoints for concentration and temperature. The control action is the coolant temperature T_c, obtained from a normalized action $a \in [-1,1]$:

$$T_c = T_c^{\min} + \frac{a+1}{2}(T_c^{\max} - T_c^{\min}) \tag{13}$$

2.6.2 *Operating Conditions and Constraints*

The reactor exhibits open-loop instability, with thermal runaway above ~ 400 K. Safety constraints include soft and hard temperature limits and bounded control inputs (Table 2). Stochasticity is introduced through measurement noise, uncertain initial conditions, and external disturbances. Observations receive Gaussian noise with $\sigma_{C_A} = 5 \times 10^{-3}$ and $\sigma_T = 0.25$ K. Initial conditions are randomized as $C_A(0) \sim \mathcal{U}[0.6, 1.0]$ and $T(0) \sim \mathcal{U}[345, 365]$ K. Disturbances are modeled as instantaneous temperature increases of $\Delta T = 20$ K applied at a fixed time step in each episode.

Table 2: Operating conditions and safety limits

Variable	Description	Value
T_{sp}	Temperature setpoint	360 K
$C_{A,sp}$	Concentration setpoint	0.35 mol/L
T_{soft}	Soft safety limit	385 K
T_{hard}	Hard safety limit	400 K
$T_c \in [\cdot]$	Coolant range	[240, 380] K

2.6.3 *Reward Function*

The reward is defined as the sum of tracking and control-variation: $r = r_{\text{track}} + r_{\text{smooth}}$. Tracking is enforced via quadratic penalties:

$$r_{\text{track}} = -1.5(T - T_{sp})^2 - 1.0(C_A - C_{A,sp})^2 \tag{14}$$

Control variation is penalized by:

$$r_{smooth} = -0.05(a_t - a_{t-1})^2 \tag{15}$$

The weights prioritize temperature regulation, which is critical for safety, while maintaining concentration tracking. A smaller penalty on control variation discourages aggressive actuation without dominating the objective.

2.6.4 *Cost Function*

The cost is defined as a quadratic penalty on temperature exceeding the risk threshold T_{soft}:

$$cost = 0.2 \max\left(0,\ T_{\text{next}} - T_{\text{soft}}\right)^2 \tag{16}$$

Episodes are terminated if numerical divergence occurs or if the hard temperature limit T_{hard} is exceeded.

2.6.5 *Training Protocol*

All agents are trained for 6×10^6 steps using identical architectures and hyperparameters (Table 3) across 9 random seeds. Disturbance was uniformly randomized per episode over steps 40–60. For CMDP-based methods, a cost limit $d = 80$ is enforced. Given the quadratic cost definition, this bounds the cumulative magnitude and duration of temperature excursions above T_{soft}. Reward, cost, and observation normalization are applied. For visualization, curves are smoothed using a moving-average window of size 100, applied per seed before computing statistics.

Table 3: Training configuration

Parameter	Value	Description
Steps per epoch	5000	On-policy rollout size
Batch size	128	SGD minibatch size
Discount factor	0.995	Reward and cost
GAE parameter	0.97	Advantage estimation
Clip ratio	0.15	PPO-based methods
Hidden layers	[32, 32]	Actor/Critic
Activation	tanh	Nonlinearity
Seeds	9	Statistical robustness

2.6.6 *Evaluation Protocol*

Policies are evaluated via deterministic rollouts, using the mean action without exploration noise. For each algorithm, 9 seeds with 100 episodes each (horizon 300) are considered, with randomized initial conditions. The disturbance step is randomized within 60–80 to prevent overfitting. Episodes are not terminated upon constraint violations, allowing full-horizon analysis. Cumulative reward, cost, violation count, and state and control trajectories are recorded. The control signal (T_c) is smoothed using a Gaussian filter with $\sigma = 3$ for visualization.

3 Results and Discussion

3.1 *Training Results*

Figure 1 summarizes training dynamics and reward–cost trade-offs, while Table 4 reports aggregate metrics. All methods train without numerical divergence, but exhibit distinct trade-offs. Unconstrained PPO achieves the highest reward (-154.07), but produces short and highly variable episodes (73.08 ± 50.59), consistent with frequent constraint violations. Among constrained methods, PPO-Lag attains the highest reward (-263.72). PCPO achieves the lowest cost (73.66) and longest episodes (150.69), with a competitive reward (-531.39). FOCOPS and P3O yield lower rewards (-950.17, -1258.06) with similar costs (81.95, 78.15), indicating less efficient reward–cost trade-offs.

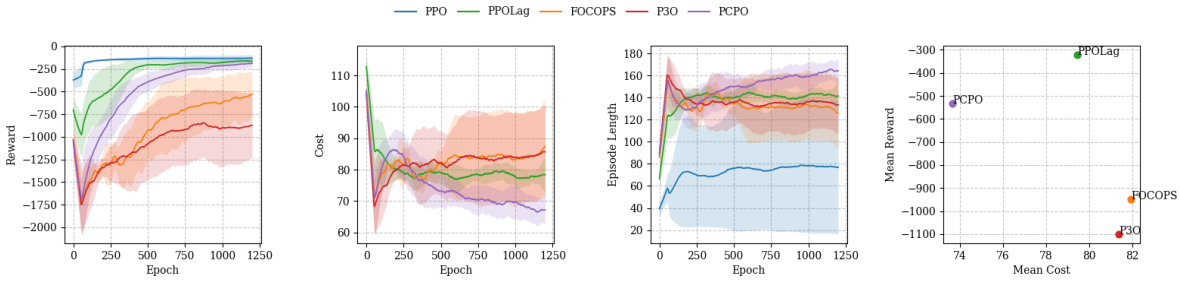

Figure 1: Training dynamics and reward–cost trade-offs.

Table 4: Performance summary (mean ± std over 3 seeds)

Algorithm	Reward (mean)	Cost (mean)	Length
PPO	**-154.07 ± 27.65**	–	73.08 ± 50.59
PPO-Lag	**-263.72 ± 72.92**	77.55 ± 1.48	147.03 ± 2.76
FOCOPS	-950.17 ± 228.76	81.95 ± 7.65	134.44 ± 17.36
P3O	-1258.06 ± 148.58	78.15 ± 1.08	142.54 ± 2.93
PCPO	-531.39 ± 105.35	**73.66 ± 3.14**	**150.69 ± 6.30**

3.2 Evaluation Results

Figure 2 shows closed-loop trajectories and reward distributions under deterministic evaluation, and Table 5 summarizes the metrics.

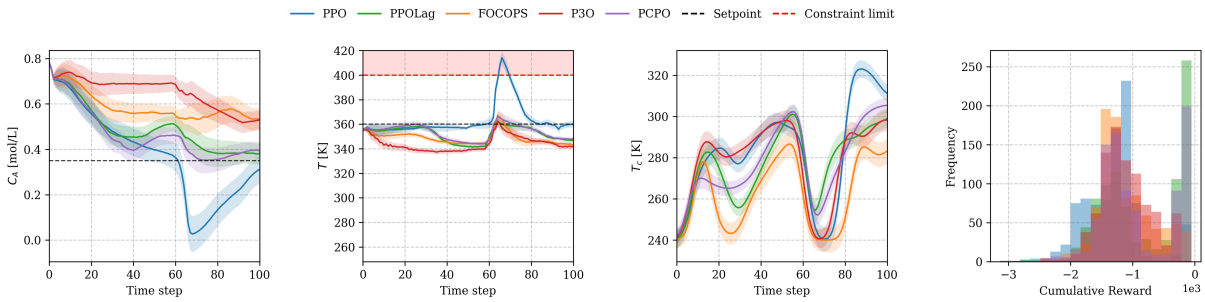

Figure 2: Closed-loop evaluation results. From left to right: C_A [mol/L], T [K], and T_c [K] as functions of time, and cumulative reward distributions. Lines denote the median across episodes, and shaded regions indicate one standard deviation. Dashed lines indicate setpoints; the red region marks the unsafe temperature range.

Table 5: Evaluation metrics (median ± std for reward/cost, mean ± std otherwise)

Algorithm	Reward	Cost	Violation Count	$\lvert\Delta T_c\rvert$
PPO	-1236.59 ± 185.56	–	8.50 ± 0.39	**18.26 ± 1.74**
PPO-Lag	**-1165.58 ± 96.16**	**214.05 ± 58.56**	**3.14 ± 0.47**	49.51 ± 2.79
FOCOPS	-1947.32 ± 85.50	385.59 ± 50.86	4.38 ± 0.47	79.29 ± 3.18
P3O	-2104.89 ± 92.27	297.02 ± 50.28	3.73 ± 0.45	70.74 ± 4.06
PCPO	-1219.85 ± 93.86	283.15 ± 58.53	4.09 ± 0.52	**25.89 ± 1.94**

Unconstrained PPO exhibits the highest violation count (8.50), reflecting poor constraint satisfaction after the disturbance. PPO-Lag achieves the best overall performance, with the highest reward (−1165.58), lowest

cost (214.05), and fewest violations (3.14), at the expense of higher control variation (49.51). PCPO provides a balanced trade-off, combining competitive reward (−1219.85), moderate cost (283.15), and reduced control effort (25.89). FOCOPS and P3O are dominated in both reward and cost (−1947.32, −2104.89; 385.59, 297.02) and exhibit larger control variation (79.29, 70.74), indicating less efficient performance.

4 Conclusions

This work evaluated state-of-the-art Safe RL methods on an unstable CSTR benchmark. In contrast to unconstrained PPO, all Safe RL algorithms significantly reduced constraint violations, confirming the effectiveness of constraint-aware formulations. However, a trade-off between performance and safety persists: prioritizing reward degrades constraint satisfaction, while emphasizing cost leads to more conservative policies with reduced performance. This sensitivity arises from the underlying CMDP structure, which requires balancing two competing objectives and introduces additional hyperparameters. Among the evaluated methods, PPO-Lag achieves the best overall performance, with the highest reward, lowest cost, and fewest constraint violations, while PCPO provides a balanced and stable safety–performance trade-off with reduced control effort.

Future work should focus on reducing this sensitivity and improving reliability, for instance, through adaptive constraint handling, improved reward–cost shaping, or hybrid approaches integrating model-based control. Overall, the results highlight important limitations of current Safe RL approaches in unstable systems, emphasizing the need for approaches that explicitly incorporate stability and safety guarantees during learning.

Acknowledgements: José R. Torraca reports financial support provided by CAPES, Finance Code 001, and Petrobras (Cooperation term 0050.0124501.23.9). Argimiro R. Secchi is grateful for financial support from CNPq (Grants No. 300744/2025-0). Other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Annaswamy, A. M. (2023). Adaptive control and intersections with reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, *6*(1), 65–93. https://doi.org/10.1146/annurev-control-062922-090153

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, *5*(1), 411–444. https://doi.org/10.1146/annurev-control-042920-020211

Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., & Knoll, A. (2024). A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(12), 11216–11235. https://doi.org/10.1109/tpami.2024.3457538

Ji, J., Zhou, J., Zhang, B., Dai, J., Pan, X., Sun, R., Huang, W., Geng, Y., Liu, M., & Yang, Y. (2024). Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, *25*(285), 1–6. http://jmlr.org/papers/v25/23-0681.html

Ray, A., Achiam, J., & Amodei, D. (2019). Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., & Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. https://doi.org/10.48550/ARXIV.2407.17032

Yang, T.-Y., Rosca, J., Narasimhan, K., & Ramadge, P. J. (2020). Projection-based constrained policy optimization. https://doi.org/10.48550/ARXIV.2010.03152

Zhang, L., Shen, L., Yang, L., Chen, S., Wang, X., Yuan, B., & Tao, D. (2022, July). Penalized proximal policy optimization for safe reinforcement learning [Main Track]. In L. D. Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22* (pp. 3744–3750). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2022/520

Zhang, Y., Vuong, Q., & Ross, K. W. (2020). First order constrained optimization in policy space. https://doi.org/10.48550/ARXIV.2002.06506