Improving Safety Filter Integration for Enhanced Reinforcement Learning in Robotics

Federico Pizarro Bejarano[1], Lukas Brunke[1,2], and Angela P. Schoellig[1,2]

***Abstract*— Reinforcement learning (RL) controllers are flexible and performant but rarely guarantee safety. Safety filters impart hard safety guarantees to RL controllers while maintaining flexibility. However, safety filters cause undesired behaviours due to the separation of the controller and the safety filter, degrading performance and robustness. This extended abstract unifies two complementary approaches aimed at improving the integration between the safety filter and the RL controller. The first extends the objective horizon of a safety filter to minimize corrections over a longer horizon [1]. The second incorporates safety filters into the training of RL controllers, improving sample efficiency and policy performance [2]. Together, these methods improve the training and deployment of RL controllers while guaranteeing safety.**

I. Introduction

Reinforcement learning (RL) can adapt to complex reward signals and unknown dynamics, which has led to superior performance in various domains, including robotics [3], [4]. However, RL lacks safety guarantees [5]. Safety filters can ensure that RL controllers operate safely while minimally interfering. They determine whether uncertified (i.e., potentially unsafe) controller inputs will violate the constraints [6], [7]. If so, the filter determines the minimal deviation from the input that results in constraint satisfaction. However, adding a safety filter changes how the controller interacts with the environment.

Contributions: This extended abstract combines insights from two works that address these challenges. The first work [1] introduces a generalization of the standard safety filter objective function, which minimizes the corrections over a horizon. This extension enables safety filters to anticipate and avoid unsafe actions more effectively, significantly reducing chattering [1], [8] (see Fig. 1). The second work [2] modifies the training process of the RL controller using safety filters. The modifications significantly improve sample efficiency, eliminate constraint violations during training, improve final performance, and reduce chattering [2], [9] (see Fig. 2). Both studies use model predictive safety filters (MPSFs) [10] but can be extended to other filters. These studies further leverage safety filters to achieve safe and efficient RL in robotics.

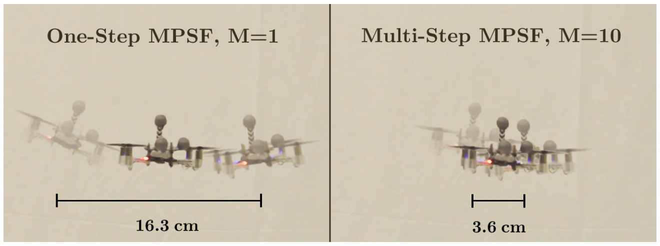

Fig. 1: Chattering caused by the standard one-step MPSF versus the multi-step MPSF [1]. The multi-step filter reduces the peak-to-peak amplitude of chattering from 16.3 cm to 3.6 cm.

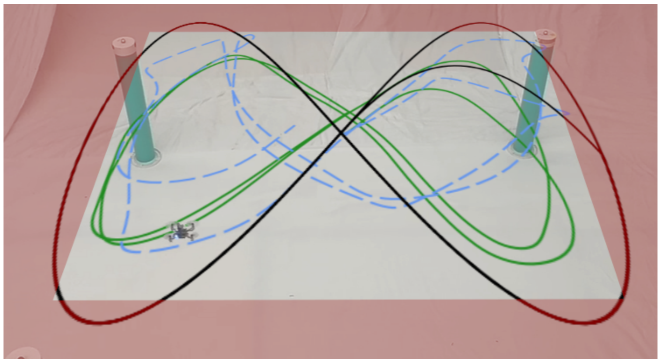

Fig. 2: An RL controller trained without a safety filter (blue) tracks a reference trajectory (black), but unforeseen interactions with the safety filter cause poor tracking. When trained with a safety filter (green), the behaviour is smoother and more performant [2]. The constraints are in red.

II. Methods

A. Multi-Step Objective Function [1]

The standard (one-step) safety filter objective function is:

$$J_{\mathrm{SF},1} = \|\pi_{\text{uncert}}(\mathbf{x}_k) - \mathbf{u}_{0|k}\|^2, \tag{1}$$

where $\mathbf{x}_k$ is the state at time step k, π_{uncert} is the RL policy, and $\mathbf{u}_{0|k}$ is the input to be applied (the optimization variable) [5]. By generalizing to multiple steps, the filter can minimize corrections over a longer prediction horizon:

$$J_{\mathrm{SF},M} = \sum_{j=0}^{M-1} w(j)\|\pi_{\text{uncert}}(\mathbf{z}_{j|k}) - \mathbf{u}_{j|k}\|^2, \tag{2}$$

where $w(\cdot) : \mathbb{N}_0 \rightarrow \mathbb{R}^+$ calculates the weights associated with the j-th correction, M is the filtering horizon, $\mathbf{z}_{j|k}$ is the estimated future state at the $(k+j)$-th time step computed at time step k, and $\mathbf{u}_{j|k}$ is the input at the $(k+j)$-th time step computed at time step k. The inputs are the optimization variables. This allows the agent to proactively correct actions to avoid unsafe states.

[1] The authors are with the Learning Systems and Robotics Lab (www.learnsyslab.org), University of Toronto, Canada, and affiliated with the University of Toronto Robotics Institute and the Vector Institute for Artificial Intelligence in Toronto. {federico.pizarrobejarano, lukas.brunke, angela.schoellig}@robotics.utias.utoronto.ca

[2] Lukas Brunke and Angela P. Schoellig are also with the Technical University of Munich and the Munich Institute for Robotics and Machine Intelligence (MIRMI).

This work has been supported by the Robotics Institute Germany, funded by BMBF grant 16ME0997K.

TABLE I: Results for the simulation experiments of the training modifications that incorporate a safety filter [2].

Metric	Std.	PC	SR	PC,SR	FA	FA,PC	FA,SR	FA,PC,SR
Return	200.2 ± 17.1	210.0 ± 14.4	213.3 ± 15.8	210.9 ± 14.6	206.1 ± 12.9	208.6 ± 14.2	211.3 ± 15.3	**214.1 ± 14.0**
Input rate of change $[\mathrm{m\,s^{-1}}]$	16.4 ± 17.0	6.6 ± 17.1	23.4 ± 15.3	**6.1 ± 7.6**	9.5 ± 2.4	11.1 ± 10.3	10.9 ± 10.4	7.5 ± 3.3
Training constraint violations [%]	82.8 ± 6.6	71.6 ± 3.5	73.3 ± 2.5	67.4 ± 6.9	8.3 ± 0.1	8.4 ± 0.1	0.23 ± 0.02	**0.22 ± 0.03**
Training time per step [ms]	**2.3 ± 0.3**	9.8 ± 1.1	4.6 ± 1.7	12.4 ± 3.0	10.5 ± 0.6	9.9 ± 1.1	11.7 ± 1.4	11.6 ± 1.5

B. *Training Modifications [2]*

We consider three modifications to the training of RL algorithms. These can be combined or used separately and applied to any RL controller and safety filter.

1) Filtering Training Actions: During training, the controller generates uncertified actions $\mathbf{u}_{\text{uncert},k} \in \mathbb{U}$. By applying the safety filter $\mathbf{u}_{\text{cert},k} = \pi_{\text{SF}}(\mathbf{x}_k, \mathbf{u}_{\text{uncert},k})$, safety is guaranteed during training [9].

2) Penalizing Corrections: We can penalize corrections during training to encourage the RL to execute safe actions. The magnitude of the correction measures how unsafe the action was. Thus, we penalize the reward by $\alpha\|\mathbf{u}_{\text{uncert},k} - \mathbf{u}_{\text{cert},k}\|_2^2$ [10], [11], where $\alpha > 0$ is a tuneable weight.

3) Safely Resetting the Environment: Sample efficiency can be improved by using the safety filter to avoid initiating an episode in an unsafe state [12]. We will sample $\mathbf{x}_0 \sim \mathbb{S}$, where $\mathbb{S}$ is the set of starting states, then determine the feasibility of certifying an input from that state [12]. If the safety filtering optimization is feasible, $\mathbf{x}_0$ is safe. If infeasible, another starting state is randomly generated until a feasible starting state is found.

III. Experimental Results

To determine the efficacy of the multi-step objective function and the training modifications, we ran experiments in the safe learning-based control simulation environment `safe-control-gym` [13] and on a real quadrotor, the Crazyflie 2.0. The underlying MPC is the robust nonlinear MPC in [14]. Proximal policy optimization (PPO) [15] was used as the RL controller.

A. *Deploying with the Multi-Step Objective*

The one-step and multi-step (with $M = 2, 5, 10$) MPSFs (based on the robust nonlinear MPC in [14]) were tested five times each on real Crazyflie 2.0 quadrotors. The task consisted of tracking a trajectory that went outside the position constraints. The weight function was set to $w(j) = 0.85^j$.

As seen in Fig. 3, our multi-step approach significantly reduces the norm of the rate of change of the inputs, reducing it by 80% compared to the one-step approach when $M = 10$. The maximum correction and magnitude of corrections are either maintained or decreased compared to the one-step approach, and both are decreased by over 30% when $M = 10$ [1]. This demonstrates that our multi-step approach is more effective at decreasing chattering and jerkiness than the standard one-step objective function while reducing the overall magnitude of corrections.

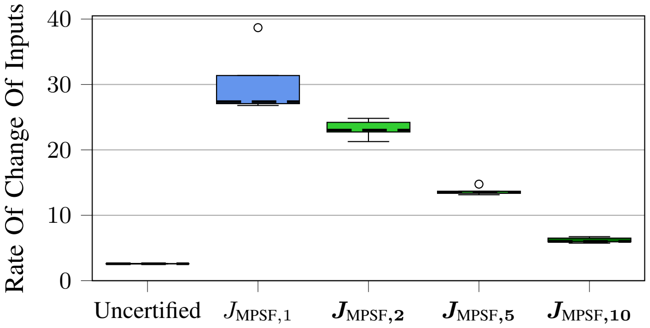

Fig. 3: The norm of the rate of change of the inputs (see [1]) for the real Crazyflie trajectory tracking experiments testing the multi-step approach. The multi-step approach significantly decreases the norm of the rate of change of the inputs, up to an 80% decrease compared to the one-step approach (in blue), without violating the constraints.

B. *Training with Safety Filters*

Every combination of the modifications was trained and evaluated. "Std." refers to the baseline with no training modifications. The other approaches are combinations of the training modifications: FA = Filtering Actions, PC = Penalizing Corrections, SR = Safe Reset.

The controllers were evaluated on a simulation of a Crazyflie 2.0 [16] using the `safe-control-gym` [13]. The trajectory tracking task consists of tracking a figure-eight reference in three dimensions. The position is constrained to be 5% smaller than the full extent of the trajectory.

From Table I, we note that penalizing the corrections reduces the number of constraint violations during training, lowers the rate of change of the inputs (see [1]), and increases the return. The safe reset modification significantly improves convergence and evaluation return. When partnered with the safe reset approach, filtering the actions reduces the constraint violations to nearly zero. Combining all the modifications leads to the best return and convergence and the least constraint violations during training [2].

IV. Conclusion

This work presents two complementary approaches to improve the integration of safety filters in reinforcement learning for robotics. The multi-step objective function effectively reduces chattering and jerkiness caused by safety filters while minimally intervening. The training modifications improve the convergence and performance of the RL agent while eliminating training-time constraint violations. Together, these approaches address key challenges in safe RL and safety filters, paving the way for broader adoption of RL in safety-critical robotic systems.

REFERENCES

[1] F. Pizarro Bejarano, L. Brunke, and A. P. Schoellig, “Multi-step model predictive safety filters: Reducing chattering by increasing the prediction horizon,” in *IEEE Conference on Decision and Control*, 2023.

[2] ——, “Safety filtering while training: Improving the performance and sample efficiency of reinforcement learning agents,” *IEEE Robotics and Automation Letters*, 2024.

[3] D. Silver *et al.*, “Mastering the game of go with deep neural networks and tree search,” *Nature*, 2016.

[4] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, “Reaching the limit in autonomous racing: Optimal control versus reinforcement learning,” *Science Robotics*, 2023.

[5] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, “Safe learning in robotics: From learning-based control to safe reinforcement learning,” *Annual Review of Control, Robotics, and Autonomous Systems*, 2022.

[6] K.-C. Hsu, H. Hu, and J. F. Fisac, “The safety filter: A unified view of safety-critical control in autonomous systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, 2023.

[7] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, “Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems,” *IEEE Control Systems Magazine*, 2023.

[8] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in *IEEE Conference on Decision and Control*, 2018.

[9] H. Krasowski, J. Thumm, M. Müller, L. Schäfer, X. Wang, and M. Althoff, “Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking,” *Transactions on Machine Learning Research*, 2023.

[10] K. P. Wabersich and M. N. Zeilinger, “A predictive safety filter for learning-based control of constrained nonlinear dynamical systems,” *Automatica*, 2021.

[11] X. Wang, “Ensuring safety of learning-based motion planners using control barrier functions,” *IEEE Robotics and Automation Letters*, 2022.

[12] X. Wang and M. Althoff, “Safe reinforcement learning for automated vehicles via online reachability analysis,” *IEEE Transactions on Intelligent Vehicles*, 2023.

[13] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, “safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics,” *IEEE Robotics and Automation Letters*, 2022.

[14] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, “A computationally efficient robust model predictive control framework for uncertain nonlinear systems – extended version,” *IEEE Transactions on Automatic Control*, 2021.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv:1707.06347*, 2017.

[16] C. E. Luis and J. L. Ny, “Design of a trajectory tracking controller for a nanoquadcopter,” Technical Report, École Polytechnique de Montréal, 2016.