Delft University of Technology

Delft Center for Systems and Control

Technical report 25-019

Probabilistically Safe and Efficient Model-Based Reinforcement Learning*

F. Airaldi, B. De Schutter, and A. Dabiri

To cite this work, please refer to the published version:

F. Airaldi, B. De Schutter, and A. Dabiri, "Probabilistically safe and efficient model-based reinforcement learning," *Proceedings of the 64th IEEE Conference on Decision and Control*, Rio de Janeiro, Brazil, pp. 5853–5860, Dec. 2025. doi:10.1109/CDC57313.2025.11312525

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://dspub.eu/25-019

Probabilistically safe and efficient model-based reinforcement learning

Filippo Airaldi, Bart De Schutter, and Azita Dabiri

***Abstract*— This paper proposes tackling safety-critical stochastic Reinforcement Learning (RL) tasks with a sample-based, model-based approach. At the core of the method lies a Model Predictive Control (MPC) scheme that acts as function approximation, providing a model-based predictive control policy. To ensure safety, a probabilistic Control Barrier Function (CBF) is integrated into the MPC controller. To approximate the effects of stochasticies in the optimal control formulation and to fulfil the probabilistic CBF condition, a sample-based approach with guarantees is employed. Furthermore, to counterbalance the additional computational burden due to sampling, a learnable terminal cost formulation is included in the MPC objective. An RL algorithm is deployed to learn both the terminal cost and the CBF constraint. Results from a numerical experiment on a constrained LTI problem corroborate the effectiveness of the proposed methodology in reducing computation time while preserving control performance and safety.**

I. INTRODUCTION

Reinforcement Learning (RL) has emerged as a successful methodology for solving complex optimal control problems, including when dealing with systems subject to uncertainty and stochastic disturbances [1]. However, employing RL in safety-critical scenarios remains in general challenging due to the inherent trial-and-error nature of the learning process and the difficulties in explicitly ensuring constraint satisfaction throughout training, even if probabilistically.

Control Barrier Functions (CBFs) have gained significant traction as an effective tool for handling safety constraints in control problems [2]. CBFs can enforce forward invariance of a safe set, thus guaranteeing safety conditions over the controlled trajectories, via an energy-based argumentation rather than relying on explicit set computations. Robust and stochastic extensions have also spun off, e.g., [3], [4], which account for uncertainties and/or disturbances affecting the system. At the same time, CBFs have been successfully integrated with various control architectures, including optimisation-based control schemes such as Model Predictive Control (MPC) [5], [6], [7], [8]. While CBFs offer guarantees on safety, their usage introduces some challenges. One of these lies in properly calibrating the CBF parameters, particularly its class $\mathcal{K}$ function. Poor tuning can severely impact the feasibility of the control problem and its closed-loop performance. Selecting an appropriate class $\mathcal{K}$ function thus involves a non-trivial trade-off between conservativeness (safety) and control performance. In this regard, adaptive formulations have been proposed in the literature that, e.g., employ auxiliary constructions [9] or leverage intelligent decision-making [7], [10] to adjust the barrier parameters autonomously.

Recently, MPC has been proposed as a promising function approximation strategy for RL algorithms, where the predictive controller acts both as policy provider and value function approximation for the underlying RL task [11], [12]. In contrast to model-free approaches, this method often results in higher sample efficiency, better interpretability and certifiability, since the MPC controller can explicitly incorporate system dynamics and systematically handle constraints. Importantly, variants of the nominal MPC formulation can also address robust and/or stochastic control problems characterised by uncertainties and/or disturbances [13], [14], [15]. Despite its benefits, the application of stochastic MPC, particularly in its sample-based forms, is often computationally demanding. One common approach to mitigate this computational complexity is to shorten the MPC prediction horizon. However, doing so can adversely affect control performance and safety due to the induced myopia of the controller. This issue is commonly addressed by introducing a terminal cost approximation, crafted to appropriately approximate the true (generally unknown) cost-to-go beyond the shortened horizon [16], [17], [18], [19]. Nonetheless, similar to the class $\mathcal{K}$ functions in CBFs, manually selecting or designing an effective terminal cost approximation introduces another trade-off between computational complexity, safety, and control performance.

In this paper, we propose a novel approach that leverages MPC-based RL combined with probabilistic CBFs and terminal cost approximation to automatically learn from data a model-based policy that ensures probabilistic safety while maintaining computational efficiency. Our methodology integrates probabilistic CBF constraints into the MPC formulation to enforce safety despite stochastic disturbances with arbitrary probability. A sample-based approximation is introduced to render the optimisation control problem tractable. This is distinct from, e.g., MPPI [20], where sampling is exploited to compute a Monte Carlo approximation of the optimal action sequence of the control problem. To address the computational complexity introduced by the sample-based approach and additional CBF constraint, we employ a shortened MPC prediction horizon alongside a learnable terminal cost approximation, which is automatically tuned via RL. Furthermore, the class $\mathcal{K}$ function within the CBF is also learned from interaction data, eliminating manual tuning and enabling adaptivity. Algorithm 1 summarises the

This research is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101018826 - CLariNet).

The authors are with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands {f.airaldi,b.deschutter,a.dabiri}@tudelft.nl

Algorithm 1: Stochastic safe MPC-based RL. See Section III for further details on each step below.

Input: Initial MPC parameters $\boldsymbol{\theta}^0$, violation prob. ε, full and short horizons N, $\overline{N}$, number of training episodes $n_{\max}$;

Output: Learnt parametrisation $\boldsymbol{\theta}^{n_{\max}}$;

1 Select number of samples M from ε, N, $\overline{N}$ empirically or via conservative bounds (24), (26);

2 Create sampled-based MPC controller (20) with learnable parametric CBF constraints (to ensure safety) and terminal cost approximation (to compensate for $\overline{N}$);

3 **for** $i = 0, \dots, n_{\max} - 1$ **do**

4 Perform MPC closed-loop task with current parametrisation $\boldsymbol{\theta}^i$;

5 Update parametrisation to $\boldsymbol{\theta}^{i+1}$ via RL;

6 **end**

proposed approach in a compact scheme.

The main contributions of this paper can thus be summarised as follows:

1) We introduce a stochastic MPC formulation with integrated probabilistic CBF constraints, explicitly designed to handle stochasticity in safety-critical tasks.
2) We provide a computationally efficient sample-based approximation of this formulation and propose to leverage RL to automatically learn both the terminal cost approximation and the CBF class $\mathcal{K}$ function.
3) We provide probabilistic safety guarantees and illustrate the effectiveness and computational advantages of our proposed method on a numerical example.

The remainder of this paper is structured as follows. In Section II, we review relevant background on safe RL, MPC as function approximation, and terminal cost approximations. Section III describes and analyses the proposed method. Simulation results validating our approach are presented in Section IV, followed by conclusions in Section V.

Notation: vector and matrix quantities are in bold. Inequalities on vectors are applied element-wise. Operation $\|\boldsymbol{y}\|_{\boldsymbol{A}}$ indicates $\sqrt{\boldsymbol{y}^\top \boldsymbol{A}\boldsymbol{y}}$, and $\boldsymbol{y} \odot \boldsymbol{z}$ the Hadamard product between $\boldsymbol{y}$ and $\boldsymbol{z}$.

II. BACKGROUND

A. Safe Reinforcement Learning

Consider the discrete-time, possibly nonlinear, stochastic system

$$\boldsymbol{s}_{t+1} = f\left(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{\omega}_t\right), \tag{1}$$

where, at each time step $t \in \mathbb{N}$, $\boldsymbol{s}_t \in \mathcal{S} \subseteq \mathbb{R}^{n_s}$ denotes its state, $\boldsymbol{a}_t \in \mathcal{A} \subset \mathbb{R}^{n_a}$ the control action, and $\boldsymbol{\omega}_t \in \Omega \subseteq \mathbb{R}^{n_d}$ the disturbance affecting the system. Dynamics $f : \mathcal{S} \times \mathcal{A} \times \Omega \to \mathcal{S}$ are assumed to be known and Lipschitz continuous w.r.t. $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$ with constant L_f over the domain $\mathcal{S} \times \mathcal{A}$.

Assumption 1 (Uncertainty)**.** *Sequences $\{\boldsymbol{\omega}_\tau\}_{\tau=t}^{t+N} \sim \mathcal{W}$, for $N > 0$, are independent and identically distributed (i.i.d.) random variables with support Ω^N. Further, a sufficient number of i.i.d. samples of these sequences can be drawn from $\mathcal{W}$ or is available (e.g., from historical data).*

For a deterministic policy $\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{A}$, parametrised in $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{n_{\boldsymbol{\theta}}}$, we define its performance as[1]

$$J\left(\pi_{\boldsymbol{\theta}}\right) \coloneqq \mathbb{E}_{\chi_{\pi_{\boldsymbol{\theta}}}}\left[\sum_{t=0}^{T} \ell\big(\boldsymbol{s}_t, \pi_{\boldsymbol{\theta}}\left(\boldsymbol{s}_t\right)\big)\right], \tag{2}$$

where $T \in \mathbb{N}$ is the horizon, $\ell : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a stage cost function, and $\chi_{\pi_{\boldsymbol{\theta}}}$ the state distribution the policy induces. In safe RL, we are primarily interested in finding a policy that optimises the performance while providing safe trajectories with high probability, i.e.,

$$\pi_{\boldsymbol{\theta}}^\star \in \arg\min_{\boldsymbol{\theta} \in \Theta} \left\{ J(\pi_{\boldsymbol{\theta}}) \;:\; \mathbb{P}\left[\bigcap_{t=0}^{T} \boldsymbol{s}_t \in \mathcal{C}\right] \geq 1 - \varepsilon \right\}, \tag{3}$$

where $\mathcal{C} = \{\boldsymbol{s} \in \mathcal{S} \mid h(\boldsymbol{s}) \geq 0\}$ denotes the desired safe set, defined by $h : \mathcal{S} \to \mathbb{R}$, a Lipschitz continuous function in $\mathcal{S}$ with constant L_h, and $\varepsilon \in (0, 1)$ the confidence level for the joint chance constraint. Note that, due to the presence of stochastic disturbances, also the state becomes a random variable and generally cannot be constrained to satisfy h with unit probability without further assumptions (e.g., boundedness of the support of $\boldsymbol{\omega}_t$).

The familiar notions of state- and action-value functions [1] apply here as well:

$$V_{\boldsymbol{\theta}}(\boldsymbol{s}_t) = \mathbb{E}_{\chi_{\pi_{\boldsymbol{\theta}}}}\left[\sum_{\tau=t}^{T} \ell\big(\boldsymbol{s}_\tau, \pi_{\boldsymbol{\theta}}\left(\boldsymbol{s}_\tau\right)\big)\right], \tag{4}$$

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \ell(\boldsymbol{s}_t, \boldsymbol{a}_t) + \mathbb{E}_{\boldsymbol{\omega}_t}\big[V_{\boldsymbol{\theta}}(\boldsymbol{s}_{t+1})\big]. \tag{5}$$

B. MPC as Function Approximation in RL

To parametrise the policy $\pi_{\boldsymbol{\theta}}$ and deploy an RL agent, (deep) neural networks are oftentimes the most common choice [21]. However, model-free approaches generally suffer from several drawbacks, as discussed in Section I. In this work, a model-based solution to (3), which leverages MPC as the function approximation scheme, is instead pursued.

Given the current state $\boldsymbol{s}_t$, consider the MPC scheme

$$\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \quad \lambda_{\boldsymbol{\theta}}(\boldsymbol{x}_0) + \mathbb{E}\left[\sum_{k=0}^{N-1} \ell_{\boldsymbol{\theta}}(\boldsymbol{x}_k, \boldsymbol{u}_k) + V_{\boldsymbol{\theta}}^{\mathrm{f}}(\boldsymbol{x}_N)\right] \tag{6a}$$

$$\text{s.t.} \quad \boldsymbol{u}_k \in \mathcal{A}, \qquad k = 0, \dots, N-1, \tag{6b}$$

$$\boldsymbol{x}_0 = \boldsymbol{s}_t, \tag{6c}$$

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\omega}_k), \; k = 0, \dots, N-1, \tag{6d}$$

$$\{\boldsymbol{\omega}_k\}_{k=0}^{N-1} \sim \mathcal{W},$$

$$\mathbb{P}\left[\bigcap_{k=1}^{N} \boldsymbol{x}_k \in \mathcal{C}\right] \geq 1 - \varepsilon, \tag{6e}$$

[1]For simplicity, we address in this paper the finite-horizon undiscounted setting, but our results can be easily extended to the infinite-horizon discounted case.

where $N \geq 0$ is the prediction horizon, $\ell_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\lambda_{\boldsymbol{\theta}}, V^{\mathrm{f}}_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ the stage, initial, and final cost approximations respectively. This scheme serves as the approximation of the value function as

$$V_{\boldsymbol{\theta}}(\boldsymbol{s}_t) = \min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \{(6a) \ : \ (6b) - (6e)\} \tag{7}$$

and it satisfies the Bellman equations so that

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \{(6a) \ : \ (6b) - (6e), \, \boldsymbol{u}_0 = \boldsymbol{a}_t\}, \tag{8}$$

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{s}_t) = \boldsymbol{u}_0^\star = \arg\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \{(6a) \ : \ (6b) - (6e)\}. \tag{9}$$

It was first shown in [11] that the solution to an MPC optimisation problem can approximate the optimal value function. This is especially intuitive if the MPC horizon N were to approach the task horizon T and we would take $\ell_{\boldsymbol{\theta}} = \ell$ and $\lambda_{\boldsymbol{\theta}}, V^{\mathrm{f}}_{\boldsymbol{\theta}} = 0$. However, in general, long prediction horizons and the stochastic arguments in (6) massively hinder the tractability of the MPC problem. In Section III, we present our approach to circumvent both issues in the context of safe RL. While in general, in addition to $V^{\mathrm{f}}_{\boldsymbol{\theta}}$, it is beneficial to have both $\ell_{\boldsymbol{\theta}}$ and $\lambda_{\boldsymbol{\theta}}$ parametrised to increase the number of degrees of freedom of the approximation scheme [11], in what follows we propose to only focus on $V^{\mathrm{f}}_{\boldsymbol{\theta}}$ for computational relief and control performance.

C. Cost-to-go Approximation

A proper choice of terminal cost $V^{\mathrm{f}}_{\boldsymbol{\theta}}$ in (6a) is essential in capturing the cost-to-go for the terminal state $\boldsymbol{x}_N$. In general, analytical forms of the true cost-to-go are often unavailable, and approximations must be used instead. As discussed in Section I, the control literature offers various solutions to this challenge. In particular, in this work, we highlight the following approaches from the literature.

1) Nonconvex Case: for a nonconvex system and stage cost, the cost-to-go approximation can be parametrised as in [18]:

$$\boldsymbol{P}_{\boldsymbol{\theta}}(\boldsymbol{c}) = L_{\boldsymbol{\theta}}(\boldsymbol{c}) L_{\boldsymbol{\theta}}^\top(\boldsymbol{c}), \tag{10}$$

$$V^{\mathrm{f,psd}}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{c}) = \left\| \boldsymbol{x} - \boldsymbol{x}^{\mathrm{f}}_{\boldsymbol{\theta}}(\boldsymbol{c}) \right\|^2_{\boldsymbol{P}_{\boldsymbol{\theta}}(\boldsymbol{c})}, \tag{11}$$

where $\boldsymbol{c} \in \mathbb{R}^{n_c}$ is the task-relevant context available at the current time step (which can include any information, e.g., state $\boldsymbol{x}$, previous actions, references, etc.), and both $\boldsymbol{x}^{\mathrm{f}}_{\boldsymbol{\theta}} : \mathbb{R}^{n_c} \to \mathbb{R}^{n_s}$ and $L_{\boldsymbol{\theta}} : \mathbb{R}^{n_c} \to \mathbb{R}^{n_s \times n_s}$ are represented by two neural networks (NNs), whose parameters are meant as included in $\boldsymbol{\theta}$. In particular, $L_{\boldsymbol{\theta}}(\boldsymbol{c})$ from (10) is a lower triangular matrix with only $\frac{1}{2} n_s (n_s + 1)$ free entries. This Cholesky decomposition-like form allows the approximate terminal cost to be positive semidefinite (PSD) w.r.t. $\boldsymbol{x}$ by construction. For a fixed $\boldsymbol{c}$, this makes optimising over the ensuing quadratic form relatively easy and cheap. At the same time, the quadratic form is context-dependent, meaning its value and gradient information will change from time step to time step, making the approximation also time-dependent. Lastly, the approach is quite malleable as the two NNs can be seamlessly scaled down or up as needed.

2) Convex Case: in the case of constrained linear time-invariant systems with quadratic regulation cost, it is well-known that the optimal value function is convex piecewise quadratic (PWQ) [22]. This result can also be extended to the stochastic setting with zero-mean, time-uncorrelated Gaussian disturbances [23]. In such a case where the value function is known to have (exactly or even approximately) a convex PWQ shape, [19] suggests the use of the approximation

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \mathrm{ReLU}\left(\boldsymbol{W}_{\boldsymbol{\theta}} \boldsymbol{x} + \boldsymbol{b}_{\boldsymbol{\theta}}\right), \tag{12}$$

$$V^{\mathrm{f,pwq}}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}_{\boldsymbol{\theta}}^\top \big(\boldsymbol{\varphi}(\boldsymbol{x}) \odot \boldsymbol{\varphi}(\boldsymbol{x})\big), \tag{13}$$

where $\boldsymbol{W}_{\boldsymbol{\theta}} \in \mathbb{R}^{m \times n_s}$, $\boldsymbol{b}_{\boldsymbol{\theta}} \in \mathbb{R}^m_{\leq 0}$ and $\boldsymbol{w}_{\boldsymbol{\theta}} \in \mathbb{R}^m_{\geq 0}$ are the adjustable weights and biases of the NN, and $\boldsymbol{\varphi} \in \mathbb{R}^m$ its hidden features. By enforcing $\boldsymbol{b}_{\boldsymbol{\theta}} < 0$ and $\boldsymbol{w}_{\boldsymbol{\theta}} \geq 0$, it is shown in [19] that this function is PWQ and convex w.r.t. $\boldsymbol{x}$. The advantage of this approximation lies in its scalability (by appropriately selecting the hidden size m) and ability to represent any PWQ convex functions by construction.

III. METHODOLOGY

This section introduces a sample-based approximation to the stochastic MPC problem. Similarly to, e.g., [18], [19], we propose to employ a learnable terminal cost formulation, coupled with a short prediction horizon, to mitigate the computational complexity induced by the sampling approach. At the same time, to preserve the probabilistic safety of the closed-loop state trajectories despite the increased myopia of the controller, we leverage the notion of CBF to guarantee step-wise forward invariance of the safe set with high probability. We adopt RL to perform training of both the terminal cost and the CBF class $\mathcal{K}$ function in an end-to-end fashion.

A. Probabilistic Control Barrier Function

In this work, we propose to leverage the CBF framework to guarantee safety. Again, it is essential to remark that, due to the stochasticity affecting the system, in general safety cannot be guaranteed with unit probability. Rather, we will take a probabilistic approach.

Definition 1 (N-Step ε-Control Invariant Set [24])**.** *A set $\mathcal{Q} \subseteq \mathbb{R}^{n_s}$ is N-step ε-control invariant for system* (1) *if, for any $\boldsymbol{s}_t \in \mathcal{Q}$, there exists a control policy such that*

$$\mathbb{P}\left[\bigcap_{\tau=1}^{N} \boldsymbol{s}_{t+\tau} \in \mathcal{Q}\right] \geq 1 - \varepsilon. \tag{14}$$

Definition 2 (Probabilistic Control Barrier Function [8])**.** *For system* (1) *and safe set $\mathcal{C} \subseteq \mathcal{S}$, the continuous function $h : \mathcal{S} \to \mathbb{R}$ is a discrete-time probabilistic CBF if there exist a class $\mathcal{K}$ function $\alpha : [0, a) \to [0, \infty)$, $\alpha(y) \leq y$, $\forall y \geq 0$, and a control action $\boldsymbol{a}_t \in \mathcal{A}$ such that, with $\xi \in [0, 1)$, it holds that*

$$\mathbb{P}\big[h(\boldsymbol{s}_{t+1}) - h(\boldsymbol{s}_t) \geq -\alpha\big(h(\boldsymbol{s}_t)\big)\big] \geq 1 - \xi, \quad \forall t \in \mathbb{N}. \tag{15}$$

Note that the above follows straightforwardly from the standard CBF definition, on top of which the probability

operator $\mathbb{P}$ has been applied since the state is now a random variable. Now, we can state a result on the step-wise probabilistic invariance guarantee for the set $\mathcal{C}$ thanks to the CBF condition.

Theorem 1. *Given a safe set $\mathcal{C} \subseteq \mathcal{S}$ defined by the continuous function $h : \mathcal{S} \to \mathbb{R}$ and current state $\boldsymbol{s}_t \in \mathcal{C}$, if h is a discrete-time probabilistic CBF, any control policy satisfying* (15) *with $\xi \leq \frac{\varepsilon}{N}$ renders the set $\mathcal{C}$ N-step ε-control invariant.*

Proof. The proof is similar to that of [8, Theorem 2]. By complement, the joint safety condition along an N-step trajectory is satisfied as long as

$$\mathbb{P}\left[\bigcup_{\tau=1}^{N} \boldsymbol{s}_{t+\tau} \notin \mathcal{C}\right] \leq \varepsilon. \tag{16}$$

Applying the union bound, we get that

$$\mathbb{P}\left[\bigcup_{\tau=1}^{N} \boldsymbol{s}_{t+\tau} \notin \mathcal{C}\right] \leq \sum_{\tau=1}^{N} \mathbb{P}[\boldsymbol{s}_{t+\tau} \notin \mathcal{C}]. \tag{17}$$

To ensure the joint probability of violation is at most ε, it is thus sufficient to require $\sum_{\tau=1}^{N} \mathbb{P}[\boldsymbol{s}_{t+\tau} \notin \mathcal{C}] \leq \varepsilon$. The simplest choice is to allocate the risk uniformly per time step, i.e., we need to satisfy

$$\mathbb{P}[\boldsymbol{s}_{t+\tau} \in \mathcal{C}] \geq 1 - \frac{\varepsilon}{N},\ \tau = 1, \dots, N. \tag{18}$$

To achieve this, we select $\xi \leq \frac{\varepsilon}{N}$ and compute the action at each time step $t + \tau$ according to (15), so that

$$\begin{aligned} &\mathbb{P}[h(\boldsymbol{s}_{t+\tau+1}) \geq 0] \\ &\quad \geq \mathbb{P}\big[h(\boldsymbol{s}_{t+\tau+1}) \geq h(\boldsymbol{s}_{t+\tau}) - \alpha\big(h(\boldsymbol{s}_{t+\tau})\big)\big] \\ &\quad \geq 1 - \xi \geq 1 - \frac{\varepsilon}{N},\ \tau = 1, \dots, N-1, \end{aligned} \tag{19}$$

where the first inequality leverages the fact that $\boldsymbol{s}_t \in \mathcal{C} \Rightarrow h(\boldsymbol{s}_t) \geq 0$, and the property $\alpha(y) \leq y$, $\forall y \geq 0$. □

This result implies that, if the control policy acts accordingly to (15) with ξ properly selected as $\frac{\varepsilon}{N}$, the state trajectory can occasionally leave the safe set $\mathcal{C}$ but the chance of doing so is bounded by ε. Note that, while leveraging the CBF condition is beneficial to safety, there are still some open issues. In particular, finding a control input directly via (15) is in general challenging due to the probability operator: a distributional characterisation of its argument may be challenging due to the possible nonlinear nature of f, h, and/or α, and would also require exact knowledge of the distribution $\mathcal{W}$. Additionally, it is well-known that properties of the ensuing control policy (such as performance) are dependent on the selection of a proper class $\mathcal{K}$ function.

B. Sample-based MPC Approximation

We can now introduce the proposed sample-based approximation of (6). Let us introduce a shortened horizon $\overline{N} \ll N$. At time step t, assume M samples $\{\boldsymbol{\omega}_\tau^{(i)}\}_{\tau=t}^{t+\overline{N}-1}$, $i = 1, \dots, M$, are available (see Assumption 1).[2] Then, we can replace the original intractable formulation (6) with the following scheme:

$$\begin{aligned} \min_{\{\boldsymbol{u}_k\}_{k=0}^{\overline{N}-1}} \quad & \lambda_{\boldsymbol{\theta}}(\boldsymbol{s}_t) \\ & + \frac{1}{M}\sum_{i=1}^{M}\left[\sum_{k=0}^{\overline{N}-1} \ell_{\boldsymbol{\theta}}\big(\boldsymbol{x}_k^{(i)}, \boldsymbol{u}_k\big) + V_{\boldsymbol{\theta}}^{\mathrm{f}}\big(\boldsymbol{x}_{\overline{N}}^{(i)}\big)\right] && \text{(20a)} \\ \text{s.t.} \quad & \boldsymbol{u}_k \in \mathcal{A}, && k = 0, \dots, \overline{N}-1, && \text{(20b)} \\ & \boldsymbol{x}_0^{(i)} = \boldsymbol{s}_t, && i = 1, \dots, M, && \text{(20c)} \\ & \boldsymbol{x}_{k+1}^{(i)} = f\big(\boldsymbol{x}_k^{(i)}, \boldsymbol{u}_k, \boldsymbol{\omega}_k^{(i)}\big), \\ & && i = 1, \dots, M,\ k = 0, \dots, \overline{N}-1, && \text{(20d)} \\ & h\big(\boldsymbol{x}_{k+1}^{(i)}\big) - h\big(\boldsymbol{x}_k^{(i)}\big) \geq \zeta - \alpha_{\boldsymbol{\theta}}\Big(h\big(\boldsymbol{x}_k^{(i)}\big)\Big), \\ & && i = 1, \dots, M,\ k = 0, \dots, \overline{N}-1. && \text{(20e)} \end{aligned}$$

Major differences lie in the safety condition (6e) being replaced by the proposed probabilistic CBF formulation (20e), and the probabilistic operators, e.g., the expectation in (6a), by sample approximation. By selecting a (much) shorter horizon, we are able to counterbalance the increased size of the optimisation problem. However, myopic policies tend to be less safe. For this reason, we leverage the CBF to ensure control invariance. Still, to avoid jeopardising safety due to the sample-based approximation, we stress that the number of samples M must be selected in such a way to guarantee that the CBF condition is satisfied with probability $1 - \frac{\varepsilon}{N}$. In what follows, we discuss how to select M in order to achieve this, thus preserving safety with confidence ε. Note that $\zeta \geq 0$ in (20e) is a (usually small) scalar required to ensure the probabilistic guarantees discussed below. It is trivial to check that, being nonnegative, its presence does not jeopardise the CBF validity. If the problem (20) is convex, ζ can be freely set to zero; for the generic nonconvex case, $\zeta \neq 0$ (see proof of Theorem 2).

Assumption 2 (Recursive Feasibility)**.** *Under the ensuing control policy, the sampled-based MPC scheme* (20) *admits a feasible solution at every time step $t \in \mathbb{N}$ almost surely.*

This assumption is a requirement for the following result, and is standard in other works, e.g., [14], [15]. At first, it might appear restrictive but in practice hard constraints are often replaced by soft constraints in stochastic/learning settings. This choice is corroborated by the probabilistic nature of the control problem, i.e., violations cannot be avoided with unit probability (without further assumptions). Furthermore, this choice is also helpful in the context of RL: during learning, it is beneficial for the RL agent to violate constraints occasionally and receive appropriate penalties so as to learn better to discern safe and unsafe behaviours [11].

Theorem 2. *Given a confidence parameter $\beta \in (0, 1)$, there exists a minimum number of samples M for which the*

[2]Since disturbances could be time-correlated, these samples must be drawn by sampling whole sequences from $\mathcal{W}$ and then considering only the first $\overline{N} - 1$ elements.

solution $\{\boldsymbol{u}_k^\star\}_{k=0}^{\overline{N}-1}$ *to the sample-based optimisation problem* (20) *satisfies*

$$\mathbb{P}\left[\bigcap_{k=0}^{\overline{N}-1} h\left(\boldsymbol{x}_{k+1}^\star\right) - h\left(\boldsymbol{x}_k^\star\right) \geq \zeta - \alpha\big(h\left(\boldsymbol{x}_k^\star\right)\big)\right] \geq 1 - \frac{\varepsilon}{N} \tag{21}$$

with probability no smaller than $1-\beta$*, where* $\boldsymbol{x}_0^\star = \boldsymbol{s}_t$ *and* $\boldsymbol{x}_{k+1}^\star = f\big(\boldsymbol{x}_k^\star, \boldsymbol{u}_k^\star, \boldsymbol{\omega}_k\big)$.

Proof. For sake of brevity, for any feasible solution to (20) we drop the implicit dependency of $\{\boldsymbol{x}_k\}_{k=0}^{\overline{N}}$ on $\{\boldsymbol{u}_k\}_{k=0}^{\overline{N}-1}$. Given $\boldsymbol{s}_t$, define the violation probability of a solution as

$$V_{\boldsymbol{u}} = \mathbb{P}\left[\bigcup_{k=0}^{\overline{N}-1} h\left(\boldsymbol{x}_{k+1}\right) - h\left(\boldsymbol{x}_k\right) < \zeta - \alpha\big(h\left(\boldsymbol{x}_k\right)\big)\right]. \tag{22}$$

Take $\xi = \frac{\varepsilon}{N}$. Analogously to Section II-C, we distinguish between two cases.

In case (20) is nonconvex, let $d_{\mathcal{A}} = \sup_{\boldsymbol{a},\boldsymbol{a}'\in\mathcal{A}} \|\boldsymbol{a}-\boldsymbol{a}'\|_\infty$ be the diameter of $\mathcal{A}$. Because $\alpha(y) \leq y$, $\forall y \geq 0$, and α is strictly increasing, α is Lipschitz continuous with constant 1. Given the Lipschitz constants of f and h, the CBF constraint (20e) is also Lipschitz continuous with constant at most $L_{\text{CBF}} = L_h L_f + L_h + L_h$. By [25, Theorem 10] we have

$$\mathbb{P}[V_{\boldsymbol{u}^\star} > \xi] \leq \left\lceil \frac{2}{\xi} \right\rceil \left\lceil \frac{2 d_{\mathcal{A}} L_{\text{CBF}}}{\zeta} \right\rceil^{\overline{N} n_a} e^{-\frac{1}{2} M \xi^2}. \tag{23}$$

By requiring that the right-hand side be $\leq \beta$, we get

$$M \geq \frac{2}{\xi^2}\left(\ln \beta^{-1} + \overline{N} n_a \ln \left\lceil \frac{2 d_{\mathcal{A}} L_{\text{CBF}}}{\zeta} \right\rceil + \ln \left\lceil \frac{2}{\xi} \right\rceil\right). \tag{24}$$

Let us tackle the special case in which (20) is convex w.r.t. its decision variables (it needs not be convex w.r.t. the disturbance). Contrarily to the previous case, here we can consider $\zeta = 0$ as it is not required. The scenario approach theory [14], [15], [26] shows that the probability of violation at the optimal solution of (20) is (possibly tightly) bounded by [26, Theorem 1]

$$\mathbb{P}[V_{\boldsymbol{u}^\star} > \xi] \leq \sum_{j=0}^{\overline{N} n_a - 1} \binom{M}{j} \xi^j \left(1-\xi\right)^{M-j}. \tag{25}$$

By requiring that the right-hand side be $\leq \beta$, we obtain

$$M \geq \frac{2}{\xi}\left(\ln \beta^{-1} + \overline{N} n_a\right). \tag{26}$$

□

Despite of arguably limited applicability, this theorem importantly confirms the intuition that, as the sample size M increases, the confidence at which the safety condition is satisfied increases. Also, note that, while the horizon $\overline{N}$ has been shrunk to combat the computational complexity due to the sampling scheme, the probabilistic safety condition has been left untouched and is still imposed over the original N-step trajectory (see right-hand side of (21) where the risk of violation is allocated over N steps instead of $\overline{N}$).

C. RL Algorithm

Note that most of the major components in (20) are parametrised in $\boldsymbol{\theta}$, including the class $\mathcal{K}$ function $\alpha_{\boldsymbol{\theta}}$ and the terminal cost function $V_{\boldsymbol{\theta}}^{\text{f}}$. We propose to adjust this parametrisation in closed loop via an MPC-based RL algorithm [11]. This approach solves the original safe RL problem (3) as the safety constraint is taken into account into the MPC function approximation while the performance cost (2) is minimised by a gradient-based RL method. Among the advantages of this approach is the fact that it bypasses the need to manually craft and select the parametrised components, which are instead adjusted by RL via interactions with the environment. This encompasses the ability also to learn $\alpha_{\boldsymbol{\theta}}$, yielding an intrinsically adaptive CBF that can automatically balance the trade-off between trajectory safety and control performance.

Because we explicitly include a learnable terminal cost term in the objective, a value-based method is leveraged here. In particular, we propose the use of Q-learning [27]. Briefly, Q-learning indirectly finds the optimal policy by solving $\min_{\boldsymbol{\theta}} \mathbb{E}\Big[\|\ell(\boldsymbol{s},\boldsymbol{a}) + V_{\boldsymbol{\theta}}(\boldsymbol{s}_+) - Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})\|^2\Big]$, where $V_{\boldsymbol{\theta}}$ and $Q_{\boldsymbol{\theta}}$ are defined in (7) and (8). The problem can be minimised via, e.g., gradient descent updates

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \delta \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}), \tag{27}$$

with $\eta > 0$ a properly selected learning rate and δ the temporal difference error. For the computation of $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$, while not straightforward, nonlinear sensitivity analysis of the MPC scheme (20) shows that this sensitivity coincides with the partial derivative of the Lagrangian w.r.t. $\boldsymbol{\theta}$ at the optimal primal-dual solution [28]. Details on the implementation can be found in, e.g., [29], [30].

IV. NUMERICAL EXPERIMENT

In this section, we test the proposed methodology on a numerical case. The experiment was implemented in Python 3.12.6 and conducted on a server with 16 AMD EPYC 7252 (3.1 GHz) processors and 252GB RAM. The optimisation problems were formulated with CasADi [31], and solved via Gurobi [32]. Source code and results are available in the following repository: `https://github.com/FilippoAiraldi/mpcrl-cbf`.

A. Problem Description

Consider the stochastic LTI system $f(\boldsymbol{s}_t, \boldsymbol{a}_t, \omega_t) = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{B}\boldsymbol{a}_t + \boldsymbol{E}\omega_t$ with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.4 \\ -0.1 & 1 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 1 & 0.05 \\ 0.5 & 1 \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} 0.03 \\ 0.01 \end{bmatrix}, \tag{28}$$

where the disturbances are time-uncorrelated zero-mean normally distributed, i.e., $\mathbb{E}[\omega_t] = 0$ and $\mathbb{E}[\omega_i \omega_j] \propto \delta(i-j)$. The control space is $\mathcal{A} = \big\{\boldsymbol{a} \in \mathbb{R}^2 : \|\boldsymbol{a}\|_\infty \leq 0.5\big\}$. The safe set is defined as $\mathcal{C} = \big\{\boldsymbol{s} \in \mathbb{R}^2 : \|\boldsymbol{s}\|_\infty \leq 3\big\}$, where the infinity-norm state constraint is turned into four separate CBFs $h_j : \mathbb{R}^2 \to \mathbb{R}$, $j = 1, \dots, 4$, i.e., $h_{2i-1}(\boldsymbol{s}) = 3 - s_i$ and $h_{2i}(\boldsymbol{s}) = 3 + s_i$, $i = 1, 2$. The RL stage cost includes

quadratic terms alongside penalties for the violation of the safety condition:

$$\ell(\boldsymbol{s}, \boldsymbol{a}) = \|\boldsymbol{s}\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}\|_{\boldsymbol{R}}^2 - c \sum_{j=1}^{4} \min\left\{0, h_j(\boldsymbol{s})\right\}, \quad (29)$$

with $\boldsymbol{Q} = \boldsymbol{I}_{2\times 2}$, $\boldsymbol{R} = 0.1\boldsymbol{I}_{2\times 2}$, and $c = 10^3$. The length of a single episode is set to $T = 30$ time steps.

B. MPC and RL Implementation

Given the current state $\boldsymbol{s}_t$, the following unit-horizon MPC scheme derived from (6) and (20) is employed as function approximation with $M = 32$ samples[3]:

$$\min_{\boldsymbol{u}_0, \Sigma} \quad \ell(\boldsymbol{s}_t, \boldsymbol{u}_0) + \frac{1}{M} \sum_{i=1}^{M} \left[c \sum_{j=1}^{4} \sigma_j^{(i)} + V_{\boldsymbol{\theta}}^{\text{f,pwq}}\left(\boldsymbol{x}_1^{(i)}\right) \right] \quad (30a)$$

$$\text{s.t.} \quad -0.5 \leq \boldsymbol{u}_0 \leq 0.5, \quad (30b)$$

$$\boldsymbol{x}_1^{(i)} = f\left(\boldsymbol{s}_t, \boldsymbol{u}_0, \omega_0^{(i)}\right), \qquad i = 1, \ldots, M, \quad (30c)$$

$$h_j\left(\boldsymbol{x}_1^{(i)}\right) - (1 - \gamma_{\boldsymbol{\theta},j})\, h_j(\boldsymbol{s}_t) + \sigma_j^{(i)} \geq 0, \quad j = 1, \ldots, 4,\ i = 1, \ldots, M, \quad (30d)$$

$$\sigma_j^{(i)} \geq 0, \qquad j = 1, \ldots, 4,\ i = 1, \ldots, M. \quad (30e)$$

As terminal cost approximation in (30a), the convex PWQ function $V_{\boldsymbol{\theta}}^{\text{f,pwq}}$ is employed with a hidden size of 16 neurons. For each CBF constraint (30d), the corresponding class $\mathcal{K}$ function is parametrised linearly, i.e., $\alpha_{\boldsymbol{\theta},j}(y) = \gamma_{\boldsymbol{\theta},j} y$, $j = 1, \ldots, 4$, where $\gamma_{\boldsymbol{\theta},j} \in [0, 1]$ is an adjustable scalar value. The whole MPC learnable parametrisation is therefore

$$\boldsymbol{\theta} = \left(\boldsymbol{W}_{\boldsymbol{\theta}}, \boldsymbol{b}_{\boldsymbol{\theta}}, \boldsymbol{w}_{\boldsymbol{\theta}}, \gamma_{\boldsymbol{\theta},1}, \ldots, \gamma_{\boldsymbol{\theta},4}\right), \quad (31)$$

where $\boldsymbol{W}_{\boldsymbol{\theta}}$, $\boldsymbol{b}_{\boldsymbol{\theta}}$ and $\boldsymbol{w}_{\boldsymbol{\theta}}$ are defined per Section II-C.2. Note that the CBF constraints (30d) have been relaxed via slack variables $\Sigma = \left\{\sigma_j^{(i)}, i = 1, \ldots, M, j = 1, \ldots, 4\right\}$ to preserve feasibility (see Assumption 2), and, since the problem is convex, we set $\zeta = 0$.

The MPC parametrisation is initialised to uniformly random values for the PWQ NN, and to $\gamma_{\boldsymbol{\theta},j} = 0.7$ for all the CBF parameters. A Q-learning agent is trained for 1000 episodes with a learning rate of 0.005 via *rmsprop* [33]. The parametrisation is updated at the end of each episode, based on the experiences observed in the last episode. Since $\gamma_{\boldsymbol{\theta},j}$ and part of the parametrisation of the PWQ NN must be constrained, a constrained step update of $\boldsymbol{\theta}$ is performed [30]. To induce exploration in an epsilon-greedy fashion, a term $\boldsymbol{q}^\top \boldsymbol{u}_0$ is added to the objective (30a), where $\boldsymbol{q} \sim \mathcal{N}\left(\boldsymbol{0}_{2\times 2}, \rho_{\boldsymbol{q}} \boldsymbol{I}_{2\times 2}\right)$. The exploration scale $\rho_{\boldsymbol{q}}$ and probability both start at 1 but decay by a factor of 0.997 after each episode. Lastly, the training procedure is repeated for 10 differently randomly seeded agents to account for randomness.

[3]Although not selected according to (26) (which in principle provides a conservative bound on the number of samples required), this M already leads to a sufficiently safe and computationally not-too-expensive policy in our test environment.

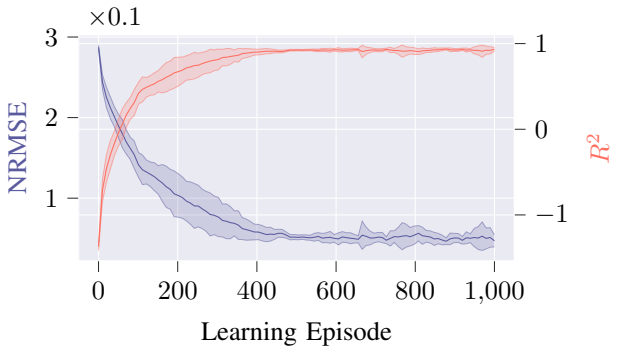

Fig. 1. Evolution of the learned terminal cost approximation in terms of normalised RMSE and coefficient of determination w.r.t. the optimal cost-to-go function for the constrained stochastic LTI experiment. Average results ± one standard deviation over 10 different seeds are reported.

C. Results

Numerical results corroborate the capability of the proposed framework in appropriately learning the terminal cost MPC component via RL, as well as the effectiveness of the learned policy compared to a full-length horizon MPC controller. Fig. 1 shows the evolution of the PWQ approximation w.r.t. the explicit optimal solution, computed in accordance to [23]. Convergence of both the normalised error and the R^2 coefficient during training provides empirical evidence that the Q-learning algorithm is able to steer the $V_{\boldsymbol{\theta}}^{\text{f}}$ term towards the real optimal one. Fig. 2 reports the evolution of the CBF parameters $\gamma_{\boldsymbol{\theta},1}$ and $\gamma_{\boldsymbol{\theta},3}$, which correspond to the lower and upper bounds on the first state. These are of more interest because, due to the dynamics, most of the violations tend to occur in these two constraints (the other two parameters $\gamma_{\boldsymbol{\theta},2}$ and $\gamma_{\boldsymbol{\theta},4}$ are omitted as they do not change as much during learning). It is important to stress again that these CBF parameters (as well as all other parameters included in $\boldsymbol{\theta}$) are adjusted by Q-learning to enhance closed-loop performance. Because constraint violations are included in the cost (29) as penalty term, safety is only indirectly taken into account by the RL algorithm. Nonetheless, since the parameters $\gamma_{\boldsymbol{\theta},j}$ are constrained to the interval $[0, 1]$ in each update, the CBFs remain valid throughout the learning process. As a matter of fact, during training, the MPC-based RL policy achieves a small empirical probability ($0.0942 \pm 0.00483\%$) of violating any constraint.

After the training phase, the learned MPC-based RL policy is evaluated against a full-length horizon stochastic MPC policy. The latter is similar to (30) but is fixed (i.e., it contains no learnable terms) and has a horizon of 12 (instead of 1), which was found to be sufficient to achieve the lowest closed-loop cost (see, e.g., [34], for a more thorough discussion on how to find such a horizon). The other hyperparameters, e.g., the number of samples M, are the same in both policies. For each evaluation episode, the initial conditions are drawn from the boundary of the maximal invariant set. Fig. 3 shows the outcomes of this evaluation comparison. Unsurprisingly, CPU time spent online in solving the MPC-RL policy is almost two orders of magnitude shorter than that for the fixed

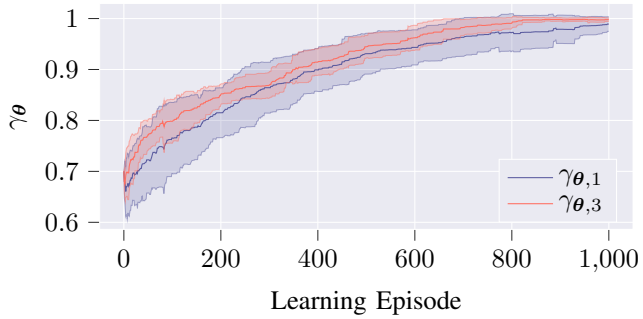

Fig. 2. Evolution of two of the linear class $\mathcal{K}$ function learnable coefficients. Average results $\pm$ one standard deviation over 10 different seeds are reported.

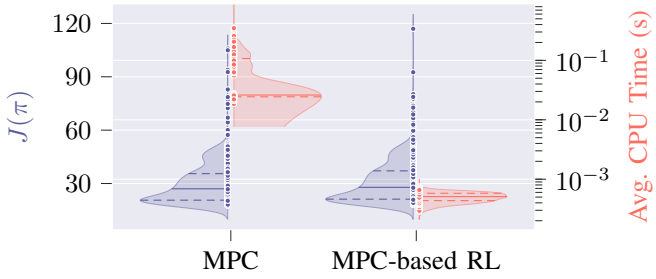

Fig. 3. Comparison between the non-learning MPC policy (horizon of 12) and the learned MPC-based RL policy (unit horizon) in terms of the total incurred cost and average solver time over different 1000 episode trials. Lines represent the second (solid) and first and third (dashed) quartiles.

MPC controller, thanks to the corresponding optimisation problem being considerably smaller. However, from the point of view of costs, both policies achieve remarkably similar closed-loop performance despite the difference in horizon lengths. This finding is further validated in Fig. 4, which reports ten state trajectories that highlight how both control policies behave rather similarly. Moreover, both policies exhibit comparable empirical constraint violation probabilities at evaluation ($0.0839 \pm 0.0138\%$ and $0.1 \pm 0.014\%$, respectively). These probabilities are also in line with the constraint violation probability recorded during training.

V. CONCLUSIONS

We have proposed a control methodology for stochastic safety-critical systems that merges MPC, CBFs and RL. The parametric MPC controller acts as the backbone, providing the control policy and value function approximation for the RL task. A probabilistic CBF formulation, integrated in the MPC scheme, is put in place to ensure safety of state trajectories with arbitrary probability. To retain tractability of the optimisation problem, the MPC horizon is (substantially) shrunk and a learnable terminal cost is introduced to combat performance drops. RL is then used to adjust the parametrisation of this learnable cost as well as the class $\mathcal{K}$ function, automatically tuning the MPC parametrisation to achieve higher closed-loop performance. A numerical

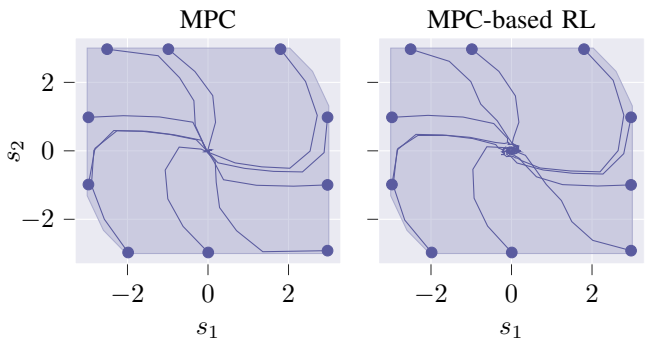

Fig. 4. Ten examples of state trajectories recorded during the evaluation of the non-learning MPC policy (horizon of 12) against the learned MPC-based RL policy (unit horizon). Also reported is the maximal invariant set.

example on a constrained LTI environment showcases the proposed method. Future work will investigate the use of more complex CBF parametrisations (e.g., neural network-based), as well as applications of the proposed methodology to nonlinear systems.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[3] A. Alan, T. G. Molnar, A. D. Ames, and G. Orosz, "Parameterized barrier functions to guarantee safety under uncertainty," *IEEE Control Systems Letters*, vol. 7, pp. 2077–2082, 2023.

[4] A. Clark, "Control barrier functions for complete and incomplete information stochastic systems," in *2019 American Control Conference (ACC)*, 2019, pp. 2928–2935.

[5] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[6] A. A. D. Nascimento, A. Papachristodoulou, and K. Margellos, "Probabilistically safe controllers based on control barrier functions and scenario model predictive control," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 1814–1819.

[7] E. Sabouni, H. M. Ahmad, V. Giammarino, C. G. Cassandras, I. C. Paschalidis, and W. Li, "Reinforcement learning-based receding horizon control using adaptive control barrier functions for safety-critical systems," in *2024 IEEE Conference on Decision and Control (CDC)*, 2024, pp. 401–406.

[8] Y. Wang, X. Shen, and H. Qian, "Stochastic model predictive control with probabilistic control barrier functions and smooth sample-based approximation," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 4798–4803.

[9] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2022.

[10] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "BarrierNet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.

[11] S. Gros and M. Zanon, "Data-driven economic NMPC using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2020.

[12] R. Reiter, J. Hoffmann, D. Reinhardt, F. Messerer, K. Baumgärtner, S. Sawant, J. Boedecker, M. Diehl, and S. Gros, "Synthesis of model predictive control and reinforcement learning: Survey and classification," *arXiv preprint arXiv:2502.02133*, 2025.

[13] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[14] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[15] M. C. Campi, S. Garatti, and M. Prandini, *Scenario optimization for MPC*. Cham. Switzerland: Springer International Publishing, 2019, pp. 445–463.

[16] N. Karnchanachari, M. de la Iglesia Valls, D. Hoeller, and M. Hutter, "Practical reinforcement learning for MPC: Learning from sparse objectives in under an hour on a real robot," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, A. M. Bayen, A. Jadbabaie, G. Pappas, P. A. Parrilo, B. Recht, C. Tomlin, and M. Zeilinger, Eds., vol. 120. PMLR, 2020, pp. 211–224.

[17] K. Seel, A. B. Kordabad, S. Gros, and J. T. Gravdahl, "Convex neural network-based cost modifications for learning model predictive control," *IEEE Open Journal of Control Systems*, vol. 1, pp. 366–379, 2022.

[18] S. Abdufattokhov, M. Zanon, and A. Bemporad, "Learning Lyapunov terminal costs from data for complexity reduction in nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 13, pp. 8676–8691, 2024.

[19] K. He, S. Shi, T. van den Boom, and B. De Schutter, "Approximate dynamic programming for constrained linear systems: A piecewise quadratic approximation approach," *Automatica*, vol. 160, p. 111456, 2024.

[20] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[21] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[22] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[23] A. E. Lim, J. B. Moore, and L. Faybusovich, "Separation theorem for linearly constrained LQG optimal control," *Systems & Control Letters*, vol. 28, no. 4, pp. 227–235, 1996.

[24] Y. Gao, K. H. Johansson, and L. Xie, "Computing probabilistic controlled invariant sets," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3138–3151, 2021.

[25] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[26] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[27] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, UK, 1989.

[28] C. Büskens and H. Maurer, "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems," in *Online Optimization of Large Scale Systems*, M. Grötschel, S. O. Krumke, and J. Rambau, Eds. Berlin, Heidelberg: Springer, 2001, pp. 3–16.

[29] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2021.

[30] F. Airaldi, B. De Schutter, and A. Dabiri, "Learning safety in model-based reinforcement learning using MPC and Gaussian processes," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5759–5764, 2023, 22nd IFAC World Congress.

[31] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[32] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2025. [Online]. Available: https://www.gurobi.com

[33] T. Tieleman, "rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, 2012.

[34] D. Chmielewski and V. Manousiouthakis, "On constrained infinite-time linear quadratic optimal control," *Systems & Control Letters*, vol. 29, no. 3, pp. 121–129, 1996.