Safe Reinforcement Learning via Projection on a Safe Set: How to Achieve Optimality?

Sebastien Gros [*] **Mario Zanon** [**] **Alberto Bemporad** [**]

[*] *Norwegian University of Technology, NTNU*
[**] *IMT School for Advanced Studies Lucca*

Abstract: For all its successes, Reinforcement Learning (RL) still struggles to deliver formal guarantees on the closed-loop behavior of the learned policy. Among other things, guaranteeing the safety of RL with respect to safety-critical systems is a very active research topic. Some recent contributions propose to rely on projections of the inputs delivered by the learned policy into a safe set, ensuring that the system safety is never jeopardized. Unfortunately, it is unclear whether this operation can be performed without disrupting the learning process. This paper addresses this issue. The problem is analysed in the context of Q-learning and policy gradient techniques. We show that the projection approach is generally disruptive in the context of Q-learning though a simple alternative solves the issue, while simple corrections can be used in the context of policy gradient methods in order to ensure that the policy gradients are unbiased. The proposed results extend to safe projections based on robust MPC techniques.

Keywords: Safe Reinforcement Learning, safe projection, robust MPC

1. INTRODUCTION

Reinforcement Learning (RL) is a tool for tackling optimal control from data. RL methods seek to increase the closed-loop performance of the control policy deployed on the system as observations are collected. RL methods often rely on Deep Neural Networks (DNN) to carry the policy approximation $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. Control policies based on DNNs provide limited opportunities for formal verifications of the resulting closed-loop behavior, and for imposing hard constraints on the evolution of the state of the real system. The development of safe RL methods is currently an open field of research (J. Garcia, 2013).

In order to tackle safety issues in RL, it has been recently proposed, see (Wabersich et al., 2019) and references therein, to use projections of the inputs delivered by the RL policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ into *safe sets*, which is known by construction to ensure the safety of the system. The construction of the safe set can, e.g., rely on specific knowledge of the system, or robust model predictive control techniques. The projection then operates as a safeguard that prevents RL from taking unsafe decisions, and adopts the safe decision that is the closest to the RL policy when RL is unsafe.

In this paper, we investigate the interaction between these safe policy projections and the learning process deployed by RL. We show that because the projection modifies the policy developed via RL, it can disrupt the learning process performed such that the learned policy can be suboptimal. The problem occurs both in the context of Q-learning and policy gradient approaches using actor-critic methods. We then propose simple techniques to alleviate the problem. In the context of Q-learning, we show that the projection technique in general jeopardizes optimality, as it is the projection of a (possibly) optimal policy on a set, and that the problem is best alleviated by relying on a direct minimization of the Q function learned by RL, under the safety constraint that the inputs must belong to the safety set, as proposed in (Zanon and Gros, 2019). In the context of the deterministic policy gradient approaches, we show that, in order to prevent the projections to bias the policy gradient estimations, the actor-critic method must be corrected with a correction which is simple to deploy. In the context of stochastic policy gradient methods, we show that the actor-critic must be constructed in a particular way to prevent the projection from biasing the policy gradient estimations. We finally show that these results extend to the case of a projection performed via robust Model Predictive Control (MPC) techniques.

The paper is structured as follows. Section 2 provides some background material. Section 3 details the projection approach in the context of Q-learning, and proposes an approach to address the resulting difficulties. Section 4 details the projection approach for policy gradient methods, both deterministic and stochastic, and proposes simple actor-critic formulations that prevent the projection from biasing the policy gradient estimations. Section 5 extends the results to the case in which the projection is performed via robust MPC. Section 6 proposes a simple simulation example using robust linear MPC in the stochastic policy gradient case, and Section 7 provides conclusions.

2. BACKGROUND

In the following, we will consider that the dynamics of the real system are possibly stochastic, evolving on continuous state-input spaces. We will furthermore consider stochastic policies π, taking the form of conditional probability densities $\pi\left[\mathbf{u}\,|\,\mathbf{x}\right] : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}_+$, denoting the probability density of selecting a given input $\mathbf{u}$ when the system is in a given state $\mathbf{x}$. We will also consider deterministic policies $\boldsymbol{\pi}\left(\mathbf{x}\right) : \mathbb{R}^n \to \mathbb{R}^m$ delivering $\mathbf{u}$ as a function of $\mathbf{x}$. For a

given stage cost $L(\mathbf{x},\mathbf{u}) \in \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and a discount factor $\gamma \in [0,1]$, the performance of a policy π is assessed via the total discounted expected cost

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{x}_k, \mathbf{u}_k) \,\middle|\, \mathbf{u}_k \sim \pi\left[\cdot \,|\, \mathbf{x}_k\right] \right], \tag{1}$$

where $\mathbb{E}_\pi$ is the expected value of the closed-loop trajectories under policy π, including the initial conditions $\mathbf{x}_0$.

In the deterministic policy case, the policy in (1) takes the form of a Dirac distribution centered at $\boldsymbol{\pi}$. The optimal policy associated to the state transition, the stage cost L and the discount factor γ is deterministic and given by

$$\boldsymbol{\pi}_\star = \arg\min_{\boldsymbol{\pi}} J(\boldsymbol{\pi}). \tag{2}$$

Reinforcement Learning seeks to find the parameters $\boldsymbol{\theta}$ such that the parametrized policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ or $\pi_{\boldsymbol{\theta}}$ approximate closely $\boldsymbol{\pi}_\star$, using observed state transitions. Q-learning methods build the optimal policy approximation indirectly, as the minimizer (Sutton and Barto, 2018):

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}) = \arg\min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{u}), \tag{3}$$

where $Q_{\boldsymbol{\theta}}$ is an approximation of the true optimal action value function $Q_\star$, solution of the Bellman equations (Bertsekas, 2007):

$$V_\star(\mathbf{x}) = \min_{\mathbf{u}} Q_\star(\mathbf{x},\mathbf{u}), \tag{4a}$$

$$Q_\star(\mathbf{x},\mathbf{u}) = L(\mathbf{x},\mathbf{u}) + \gamma \mathbb{E}\left[V_\star(\mathbf{x}_+) \,|\, \mathbf{x},\, \mathbf{u}\right]. \tag{4b}$$

The approximation $Q_{\boldsymbol{\theta}} \approx Q_\star$ is built using Temporal-Difference or Monte-Carlo techniques.

In contrast, policy gradient techniques manipulate directly the policy parameters according to the policy gradients $\nabla_{\boldsymbol{\theta}} J$ (Sutton et al., 1999). Actor-critic techniques evaluate the policy gradient resulting from a stochastic policy as (Sutton et al., 1999)

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}[\mathbf{u} \,|\, \mathbf{x}] A_{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u})\right], \tag{5}$$

where $A_{\pi_{\boldsymbol{\theta}}}$ is the advantage function associated to the policy $\pi_{\boldsymbol{\theta}}$, defined as

$$A_{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) = Q_{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) - V_{\pi_{\boldsymbol{\theta}}}(\mathbf{x}), \tag{6}$$

and where

$$V_{\pi_{\boldsymbol{\theta}}}(\mathbf{x}) = \mathbb{E}\left[L(\mathbf{x},\mathbf{u}) + \gamma V_{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_+) \,|\, \mathbf{x},\, \mathbf{u}\right], \tag{7a}$$

$$Q_{\pi_{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{u}) = L(\mathbf{x},\mathbf{u}) + \gamma \mathbb{E}\left[V_{\pi_{\boldsymbol{\theta}}}(\mathbf{x}_+) \,|\, \mathbf{x},\, \mathbf{u}\right], \tag{7b}$$

are the value and action-value functions associated to $\pi_{\boldsymbol{\theta}}$.

Similarly, the policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ associated to a deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ reads as (Silver et al., 2014)

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}) \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{x}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}))\right], \tag{8}$$

where the advantage function $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ is defined by (6)-(7) taken over a Dirac-like policy density corresponding to a deterministic policy. The advantage functions $A_{\pi_{\boldsymbol{\theta}}}$ and $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ can be estimated using Temporal-Difference or Monte-Carlo techniques.

In the context of Reinforcement-Learning, enforcing the safety of the inputs generated by a policy is not trivial (J. Garcia, 2013). Indeed, for safety-critical systems, discovering unsafe inputs from experiments is overly costly, and is typically rather done in extensive simulation campaigns. As an alternative, recent publications have proposed to approach the safety problem underlying RL by adding a safety layer to the RL process, which serves as a safeguard to the policy, see (Wabersich et al., 2019) and references therein. We detail that approach next.

2.1 Safe Policy

In this paper, we consider Reinforcement Learning subject to safety limitations. More specifically, we will consider constraints:

$$\mathbf{h}(\mathbf{x},\mathbf{u}) \leq 0 \tag{9}$$

that must be respected at all time in order for the system safety to be ensured. Moreover, we will consider a (possibly) state-dependent safe set $\mathbb{S}(\mathbf{x})$ such that

$$\mathbf{u}_k \in \mathbb{S}(\mathbf{x}_k), \qquad \forall\, k, \tag{10}$$

entails that (9) is satisfied at all times. We ought to stress here the difference between (9) and $\mathbb{S}(\mathbf{x})$. Satisfying (9) at time k entails that the system is safe at that time k, while $\mathbb{S}$ is such that enforcing (10) at time k entails that the system safety *can* be guaranteed at all time in the future. In the following, we will assume that $\mathbb{S}$ can be described via inequality constraints on $\mathbf{s}$, typically different than $\mathbf{h}$:

$$\mathbb{S}(\mathbf{x}) = \{\, \mathbf{u} \;|\; \mathbf{s}(\mathbf{x},\mathbf{u}) \leq 0 \}. \tag{11}$$

Set $\mathbb{S}(\mathbf{x})$ can be complex and non-convex. Let us additionally label $\mathbb{X}$ the set of states $\mathbf{x}$ such that $\mathbb{S}(\mathbf{x})$ is non-empty, and $\mathbb{W} = \{\, \mathbf{x},\mathbf{u} \;|\; \mathbf{s}(\mathbf{x},\mathbf{u}) \leq 0\}$. In some applications, the safe set $\mathbb{S}$ can be computed explicitly using reachability analysis, but that can be prohibitively difficult in general. Inner convex approximations can then be needed. An approach based on an implicit representation has been the object of recent publications (Zanon and Gros, 2019; Gros and Zanon, 2020).

Assuming that a safe set $\mathbb{S}$ is available, a natural approach to ensure the feasibility of a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ learned via Reinforcement Learning techniques is to perform a projection into the safe set $\mathbb{S}$, i.e., to solve online the problem:

$$\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x}) = \arg\min_{\mathbf{u}} \quad \frac{1}{2}\|\mathbf{u} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})\|^2 \tag{12a}$$

$$\text{s.t.} \quad \mathbf{s}(\mathbf{x},\mathbf{u}) \leq 0, \tag{12b}$$

hence seeking the closest safe input to the RL policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ under the Euclidian norm $\|\cdot\|$. While (12) imposes safety by construction, the optimality of the projected policy $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}$ is, in general, not guaranteed if $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is obtained via RL techniques that disregard the fact that the projection operation (12) takes place. The resulting optimality loss is arguably problem-dependent, and not investigated here. In this paper, we will focus on how (12) can be combined with RL such that optimality of $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}$ is achieved.

3. SAFE Q-LEARNING VIA PROJECTION

In this section we consider the deployment of the Q-learning technique under the safety limitation (11). The minimization in (4a) is then restricted to $\mathbb{S}(\mathbf{x})$. In the context of Q-learning, one seeks to adjust the parameters $\boldsymbol{\theta}$ supporting the function approximation $Q_{\boldsymbol{\theta}}$ such that $Q_{\boldsymbol{\theta}} \approx Q_\star$ is achieved in some sense. The parameters are typically adjusted using Temporal-Difference (TD) or Monte-Carlo techniques, aimed at (approximately) solving the least-squares problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_\star - Q_{\boldsymbol{\theta}}\right)^2\right]. \tag{13}$$

In a safe-learning context, the expected value in (13) is restricted to the safe state-input set $\mathbb{W}$, such that $Q_{\boldsymbol{\theta}} \approx Q_\star$ may only hold in $\mathbb{W}$. The RL policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is then selected

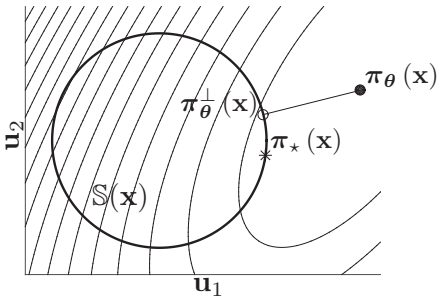

Fig. 1. Illustration of the possible loss of optimality resulting from using the projection (12) in Q-learning.

according to (3). Let us then investigate the effect of applying the projection (12) on the policy obtained from (3). To that end, let us introduce a trivial but useful result.

Lemma 1. Assume that $Q_{\boldsymbol{\theta}} = Q_\star$ holds over $\mathbb{W}$. Then the optimal policy under the safety requirement (10)-(11) is provided by:

$$\boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x}) = \arg\min_{\mathbf{u}} \quad Q_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{u}) \tag{14a}$$

$$\text{s.t.} \quad \mathbf{s}(\mathbf{x},\mathbf{u}) \leq 0, \tag{14b}$$

Proof. By contradiction. Let us assume there is a safe policy $\tilde{\boldsymbol{\pi}}_{\text{safe}}$ that achieves better closed-loop performance than $\boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})$ on $\mathbb{X}$. Because $\tilde{\boldsymbol{\pi}}_{\text{safe}}$ is safe, it follows that

$$\tilde{\boldsymbol{\pi}}_{\text{safe}}(\mathbf{x}) \in \mathbb{S}(\mathbf{x}), \qquad \forall \mathbf{x} \in \mathbb{X}. \tag{15}$$

If $\tilde{\boldsymbol{\pi}}_{\text{safe}}$ achieves better closed-loop performance than $\boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})$, and since $Q_{\boldsymbol{\theta}} = Q_\star$ holds over $\mathbb{W}$, then there is a $\mathbf{x} \in \mathbb{X}$ such that:

$$Q_\star(\mathbf{x}, \tilde{\boldsymbol{\pi}}_{\text{safe}}(\mathbf{x})) < Q_\star\left(\mathbf{x}, \boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})\right). \tag{16}$$

However, since both $\tilde{\boldsymbol{\pi}}_{\text{safe}}$ and $\boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})$ are restricted to deliver inputs in $\mathbb{S}(\mathbf{x})$, (16) is in contradiction with (14). ■

Remark 1. Note that because $Q_{\boldsymbol{\theta}} = Q_\star$ may not hold outside of $\mathbb{W}$, $Q_{\boldsymbol{\theta}}$ may take its minimum outside of $\mathbb{S}(\mathbf{x})$ for some states $\mathbf{x} \in \mathbb{X}$. As a result, constraint (14b) is required in order to generate a safe policy.

3.1 Projection Approach for Q-Learning

Consider the projection (12) of the policy (3) obtained via Q-learning. We ought to first observe that if $Q_{\boldsymbol{\theta}} = Q_\star$ holds over $\mathbb{W}$, the projected policy is optimal whenever the learned policy $\boldsymbol{\pi}_{\boldsymbol{\theta}} \in \mathbb{S}(\mathbf{x})$. Unfortunately, this observation does not necessarily extend to the situation where $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}) \notin \mathbb{S}(\mathbf{x})$. In order to support this observation, let us consider a trivial example displayed in Fig. 1. This shows that $\boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})$ does not hold in general. However, Lemma 1 readily delivers a way to alleviate this problem: assuming that a Q-function approximation $Q_{\boldsymbol{\theta}} \approx Q_\star$ over $\mathbb{Z}$ has been learned, a safe policy can be devised from using $\boldsymbol{\pi}^{\text{safe}}_\star(\mathbf{x})$ obtained from (14) as opposed to a generic projection (12). One then must be careful to include the input restriction $\mathbf{u} \in \mathbb{S}(\mathbf{x})$ in the evaluation of the TD error underlying the Q-learning. When using SARSA, no special care needs to be taken in the learning process, as (14) generates all inputs in $\mathbb{S}(\mathbf{x})$. An approach to formulate (14) via robust MPC is presented in (Zanon and Gros, 2019).

This section shows that the direct minimization (14) of the Q function approximation under the safety constraints is arguably better suited than the two steps approach: (3) followed by (12). We ought to extend the discussion to the context of the policy gradient methods using actor-critic techniques. This discussion is more technical, and is the object of the next section.

4. SAFE POLICY GRADIENT VIA PROJECTION

Policy gradient methods are often preferred over Q-learning because they alleviate the known issue that solving the least-squares problem (13) does not necessarily imply that one has found parameter $\boldsymbol{\theta}$ that yields the best closed-loop performance of the policy (3). Indeed, policy gradient methods seek a direct minimization of the closed-loop cost (1) via gradient steps over (1), and therefore yield (at least locally) optimal policy parameters. Similarly to the discussion of Section 3, when deploying policy gradient techniques jointly with a projection on the safe set (12), the optimality of the resulting policy is unclear. As a matter of fact, we will show in this section that the learning process ought to be corrected in order for the estimation of the gradient of (1) to be unbiased. Subsection 4.1 will cover the deterministic policy gradient case, while subsection 4.2 will cover the stochastic policy gradient case.

4.1 Projected Policy and Deterministic Policy Gradient

In the context of deterministic policies, we will show next that a correction must be applied in the policy gradient computation to account for the safe projection (12). This correction is provided in the following Proposition.

Proposition 1. Consider the projection (12) where $\|\cdot\|$ stands for the Euclidian norm, and assume that the constraints (12b) satisfy the Linear Constraint Qualification (LICQ) and strict Second-Order Sufficient Conditions (SOSC). The gradient of the projected policy $\boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}$ with respect to the policy parameters $\boldsymbol{\theta}$ then reads as:

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})\, M(\mathbf{x}), \tag{17a}$$

$$M(\mathbf{x}) = \mathcal{N}\left(\mathcal{N}^\top H \mathcal{N}\right)^{-1} \mathcal{N}^\top, \tag{17b}$$

where $\mathcal{N} \in \mathbb{R}^{m \times n_{\mathbb{A}}}$ is a state-dependent orthonormal null space to the gradient of the strictly active constraints, i.e.:

$$\nabla_{\mathbf{u}} \mathbf{s}_{\mathbb{A}}\left(\mathbf{x}, \boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}(\mathbf{x})\right)^\top \mathcal{N}(\mathbf{x}) = 0, \quad \mathcal{N}(\mathbf{x})^\top \mathcal{N}(\mathbf{x}) = I, \tag{18}$$

with $\mathbb{A}$ gathering the set of strictly active constraints $\mathbf{s}$, and H is the Hessian associated to (12).

Proof. The solution to (12) satisfies the KKT conditions:

$$\mathbf{r} = \begin{bmatrix} \nabla_{\mathbf{u}} \mathcal{L}(\mathbf{x},\mathbf{u},\boldsymbol{\mu}) \\ \text{diag}(\boldsymbol{\mu}_{\mathbb{A}}) \mathbf{s}_{\mathbb{A}}(\mathbf{x},\mathbf{u}) \end{bmatrix} = 0, \tag{19}$$

where $\mathcal{L} = \frac{1}{2}\|\mathbf{u} - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})\|^2 + \boldsymbol{\mu}^\top \mathbf{s}(\mathbf{x},\mathbf{u})$. The Implicit Function Theorem guarantees that if LICQ and SOSC hold, the gradient of the projected policy reads as:

$$\begin{bmatrix} H & \nabla_{\mathbf{u}} \mathbf{s}_{\mathbb{A}} \\ \nabla_{\mathbf{u}} \mathbf{s}_{\mathbb{A}}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \partial \mathbf{u}/\partial \boldsymbol{\theta} \\ \partial \boldsymbol{\mu}_{\mathbb{A}}/\partial \boldsymbol{\theta} \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}^\top \\ \mathbf{0} \end{bmatrix}. \tag{20}$$

We then observe that $\nabla_{\mathbf{u}} \mathbf{s}_{\mathbb{A}}^\top \partial \mathbf{u}/\partial \boldsymbol{\theta} = 0$ entails that

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\perp}_{\boldsymbol{\theta}}(\mathbf{x})^\top = \frac{\partial \mathbf{u}}{\partial \boldsymbol{\theta}} = \mathcal{N}\mathbf{n}, \tag{21}$$

for some vector $\mathbf{n}$. We further observe that:

$$\mathcal{N}^\top \left(H \frac{\partial \mathbf{u}}{\partial \boldsymbol{\theta}} + \nabla_{\mathbf{u}} \mathbf{s}_{\mathbb{A}} \frac{\partial \boldsymbol{\mu}_{\mathbb{A}}}{\partial \boldsymbol{\theta}} \right) = -\mathcal{N}^\top \nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}^\top$$

follows from (20), such that, using (18) and (21) we get

$$\mathcal{N}^\top H \mathcal{N} \mathbf{n} = -\mathcal{N}^\top \nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L}^\top. \tag{22}$$

Since we have

$$\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L} = -\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}, \quad H = \mathbf{I} + \nabla^2_{\mathbf{u}} \left(\boldsymbol{\mu}_{\mathbb{A}}^\top \mathbf{s}_{\mathbb{A}} \right), \tag{23}$$

this entails $\mathbf{n} = (\mathcal{N}^\top H \mathcal{N})^{-1} \mathcal{N}^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^\top$. ■

Hence the gradient of the projected policy is a form of projection of the gradient of the original policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})$ into the null-space of the safety constraints. We will define $\mathcal{N}(\mathbf{x}) = I_{m\times m}$ for all $\mathbf{x}$ for which all constraints are strictly inactive, and $\mathcal{N}(\mathbf{x}) = 0_{m\times 1}$ for all $\mathbf{x}$ where the active constraints fully block the inputs. We observe that the set of states $\mathbf{x}$ where some constraints are weakly active—such that the gradient of the policy is only defined in the sense of its sub-gradients—is of zero measure and can therefore be disregarded in the context discussed here. In the particular case of a safety set $\mathbb{S}$ described as a polytope, such that the constraints $\mathbf{s}$ are affine, $H = \mathbf{I}$ holds and matrix M simplifies to $M = \mathcal{N}\mathcal{N}^\top$.

We can then form the Corollary to Proposition 1 providing a correct policy gradient evaluation.

Corollary 1. Let us assume that (12) fulfills LICQ and SOSC. Then the policy gradient associated to the safe policy $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}$ reads as:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}\right) &= \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}\right] \\ &= \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} M \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}\right] \end{aligned} \tag{24}$$

where $A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}$ is the advantage function associated to the projected policy $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}$. All terms in (24) are evaluated at $\mathbf{x}$, $\mathbf{u} = \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x})$ with $\mathbf{x}$ distributed according to the probability density of the states in closed-loop under policy $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}$.

Proof. We observe that for any $\mathbf{x}$ such that no constraint is weakly active, the equality

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}} = \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} M \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}$$

holds. If (12) fulfills the LICQ condition, the set of states where some constraints are weakly active is of zero-measure, such that the equality

$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}\right) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}\right] = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} M \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}\right]$$

holds. ■

If deploying the projected policy approach (12) and an actor-critic method not accounting for the projection operation, the policy gradient will generally be such that:

$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}\right) \neq \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} \nabla_{\mathbf{u}} A_{\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}}\right], \tag{25}$$

where the projection matrix $M(\mathbf{x})$ is omitted. This omission will, in general, produce a biased policy gradient (25) if the policy projection is not accounted for in the RL method when computing the policy gradient. It is therefore recommended to form and use the projection matrix M when computing the policy gradient.

It can be advantageous in some cases to adopt a stochastic policy gradient method instead of the deterministic one discussed in this section. In the stochastic policy gradient, the same question arises regarding the learning process being biased by the projection in the safe set. We discuss this case in the next subsection.

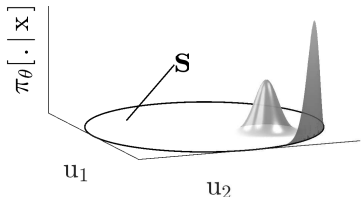

Fig. 2. Illustration of the Dirac-like effect resulting from projecting a Normally distributed stochastic policy on a safe set, chosen as a circle here.

4.2 Projected Policy and Stochastic Policy Gradient

When using a stochastic policy gradient technique, the inputs are chosen as samples $\mathbf{u}_\mathrm{s}$ drawn from a parametrized conditional probability density representing the policy:

$$\mathbf{u}_\mathrm{s} \sim \pi_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]. \tag{26}$$

The safe projection then ought to be performed over the samples $\mathbf{u}_\mathrm{s}$, i.e.:

$$\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}_\mathrm{s}) = \arg\min_{\mathbf{u}} \quad \frac{1}{2}\|\mathbf{u} - \mathbf{u}_\mathrm{s}\|^2 \tag{27a}$$

$$\text{s.t.} \quad \mathbf{s}(\mathbf{x}, \mathbf{u}) \leq 0. \tag{27b}$$

The inputs generated by $\boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}_\mathrm{s})$ are safe by construction. The resulting projected policy is itself stochastic, as it results from the nonlinear transformation (27) of the probability density $\pi_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$. Let us label the probability density resulting from the projection of the stochastic policy $\pi_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$ via (27) as $\pi^\perp_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$. Unfortunately, since the projection operator defined by (27) is not injective, the density $\pi^\perp_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$ can adopt a "Dirac-like" structure on the boundary $\partial\mathbb{S}$ of the safe set $\mathbb{S}$, due to the fact that sets of inputs of dimension larger than one is projected onto a single point on $\partial\mathbb{S}$. This issue is illustrated in Fig. 2. As a result, the score function of $\pi^\perp_{\boldsymbol{\theta}}$ is not trivially defined, and the construction of the policy gradient of $\pi^\perp_{\boldsymbol{\theta}}$ is not obvious. The following proposition shows that a trivial modification of the stochastic policy gradient allows one to circumvent this difficulty.

Proposition 2. The policy gradient associated to $\pi^\perp_{\boldsymbol{\theta}}$ is given by the actor-critic equation:

$$\nabla_{\boldsymbol{\theta}} J\left(\pi^\perp_{\boldsymbol{\theta}}\right) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}[\mathbf{u}_\mathrm{s}|\mathbf{x}] A_{\pi^\perp_{\boldsymbol{\theta}}}(\mathbf{x}, \mathbf{u})\right], \tag{28}$$

where $\mathbf{u} = \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}_\mathrm{s})$ is the input obtained from (27) satisfying LICQ and SOSC, and the expected value operator $\mathbb{E}[\cdot]$ is taken over the state and input distribution obtained in closed-loop under the projected stochastic policy $\pi^\perp_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$.

Proof. In order to build a proof using simple arguments, let us consider the interior-point approximation of the projection problem (27):

$$\boldsymbol{\pi}_\tau(\mathbf{x}, \mathbf{u}_\mathrm{s}) = \arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u} - \mathbf{u}_\mathrm{s}\|^2 - \tau \sum\nolimits_i \log\left(-\mathbf{h}_i(\mathbf{x}, \mathbf{u})\right), \tag{29}$$

such that $\left\|\boldsymbol{\pi}_\tau(\mathbf{x}, \mathbf{u}_\mathrm{s}) - \boldsymbol{\pi}^\perp_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}_\mathrm{s})\right\| = O(\tau)$ holds. Let us define $\pi^\tau[\,\cdot\,|\,\mathbf{x}\,]$ the density resulting from transforming $\pi_{\boldsymbol{\theta}}[\,\cdot\,|\,\mathbf{x}\,]$ via (29). If (27) satisfying LICQ and SOSC, then (29) is locally bijective in $\mathbf{u}_\mathrm{s}$, and the score function associated to π^τ is well-defined. The associated policy gradient reads as:

$$\nabla_{\boldsymbol{\theta}} J(\pi^\tau) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log \pi^\tau[\mathbf{u}|\mathbf{x}] A_{\pi^\tau}(\mathbf{x}, \mathbf{u})\right], \tag{30}$$

where $\mathbf{u} = \boldsymbol{\pi}_\tau(\mathbf{x}, \mathbf{u}_\mathrm{s})$. Let us further define function $\boldsymbol{\pi}_\tau^{-1}$ the local inverse of $\boldsymbol{\pi}^\tau$ at $\mathbf{u}_\mathrm{s}$, i.e.,

$$\boldsymbol{\pi}_\tau^{-1}(\mathbf{x}, \boldsymbol{\pi}_\tau(\mathbf{x}, \mathbf{u})) = \mathbf{u}, \tag{31}$$

holds in a neighborhood of $\mathbf{u}_\mathrm{s}$. The existence of (31) is guaranteed for $\tau > 0$ if (27) satisfies LICQ and SOSC. We then observe that the transformation (29) of the density $\pi_{\boldsymbol{\theta}}$ yields:

$$\pi^\tau[\mathbf{u} \,|\, \mathbf{x}] = \pi\left[\, \boldsymbol{\pi}_\tau^{-1}(\mathbf{x}, \mathbf{u}) \,|\, \mathbf{x} \,\right] \det\left(\frac{\partial \boldsymbol{\pi}_\tau^{-1}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right). \tag{32}$$

For $\mathbf{u}_\mathrm{s}$ given, (29) is independent of $\boldsymbol{\theta}$, such that

$$\nabla_{\boldsymbol{\theta}} \det\left(\frac{\partial \boldsymbol{\pi}_\tau^{-1}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\right) = 0. \tag{33}$$

As a result, the score function of π^τ reads as:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \log \pi^\tau[\mathbf{u} \,|\, \mathbf{x}] &= \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left[\, \boldsymbol{\pi}_\tau^{-1}(\mathbf{x}, \mathbf{u}) \,|\, \mathbf{x} \,\right] \\ &= \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}[\, \mathbf{u}_\mathrm{s} \,|\, \mathbf{x} \,], \end{aligned} \tag{34}$$

where $\mathbf{u}_\mathrm{s}$ is the sample corresponding to $\mathbf{u}$ obtained from (29). Combining (34) and (30), we observe that

$$\nabla_{\boldsymbol{\theta}} J(\pi^\tau) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}[\, \mathbf{u}_\mathrm{s} \,|\, \mathbf{x} \,] A_{\pi^\tau}(\mathbf{x}, \boldsymbol{\pi}_\tau(\mathbf{x}, \mathbf{u}_\mathrm{s}))\right].$$

The equivalence between (29) and (27) for $\tau \to 0$ implies that (28) holds. ■

Remark 2. Proposition 2 allows one to use the projection technique in the context of RL based on a stochastic policy approach together with an actor-critic technique, where the score function of the unprojected policy can be used in conjunction with the advantage function associated to the projected policy. The score function of the unprojected policy must then be evaluated on the unprojected sample, rather than on the projected input applied to the system.

As mentioned earlier, the construction of the constraints $\mathbf{s}$ underlying the safe set $\mathbb{S}(\mathbf{x})$ can be difficult. We extend next the proposed results to MPC-based techniques allowing one to build the safety constraints implicitly, via model predictive control techniques.

5. MPC-BASED PROJECTIONS

It is in general difficult to build the safe set $\mathbb{S}$ from condition (9). Indeed, an input $\mathbf{u}$ applied at a given time k can have lasting consequences and, while not endangering the system at time k, jeopardize its safety in the future. In order to alleviate this problem, the safety constraints can be built implicitly via Model Predictive Control (MPC) techniques. In that context, let us consider

$$\mathbf{X}_k\left(\mathbf{x}, \mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right), \quad k = 0, \ldots, \infty \tag{35}$$

an outer approximation of the trajectory dispersion of the real system starting from the initial conditions $\mathbf{x}$, hence $\mathbf{X}_0\left(\mathbf{x}, \mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right) = \mathbf{x}$, and subject to the input sequence $\mathbf{u}_0$, $\mathbf{u}_k = \boldsymbol{\pi}^{\mathbb{S}}(\mathbf{x}_k)$, where $\boldsymbol{\pi}^{\mathbb{S}}$ is an arbitrary policy. The safe set can then be described as an inner approximation:

$$\begin{aligned}\mathbb{S}(\mathbf{x}) \subseteq \big\{\, \mathbf{u}_0 \;\big|\; \exists\, \boldsymbol{\pi}^{\mathbb{S}} \quad \text{s.t.} \quad & \mathbf{h}\left(\mathbf{x}_k, \boldsymbol{\pi}^{\mathbb{S}}(\mathbf{x}_k)\right) \le 0 \\ & \forall\, \mathbf{x}_k \in \mathbf{X}_k\left(\mathbf{x}, \mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right), \quad \forall\, k > 0\big\}, \end{aligned} \tag{36}$$

which can then be used in (12) or (27). If MPC techniques are used, a generalization of the projection technique can be considered. In the deterministic policy case, one can then use the generic robust formulation:

$$\left(\mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right)(\mathbf{x}_0) = \arg\min_{\mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}} \frac{1}{2}\|\mathbf{u}_0 - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x}_0)\|^2 + \phi\left(\boldsymbol{\pi}^{\mathbb{S}}, \boldsymbol{\pi}_{\boldsymbol{\theta}}\right) \tag{37a}$$

$$\text{s.t.} \quad \mathbf{h}(\mathbf{x}, \mathbf{u}_0) \le 0, \tag{37b}$$

$$\mathbf{h}\left(\mathbf{x}_k, \boldsymbol{\pi}^{\mathbb{S}}(\mathbf{x}_k)\right) \le 0, \tag{37c}$$

$$\forall\, \mathbf{x}_k \in \mathbf{X}_k\left(\mathbf{x}, \mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right), \ \forall\, k > 0.$$

We can then select $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}(\mathbf{x}) = \mathbf{u}_0(\mathbf{x})$ as a safe control input. In the stochastic policy case, the equivalent formulation reads as:

$$\left(\mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right)(\mathbf{x}_0) = \arg\min_{\mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}} \frac{1}{2}\|\mathbf{u}_0 - \mathbf{u}_\mathrm{s}\|^2 + \phi\left(\boldsymbol{\pi}^{\mathbb{S}}, \pi_{\boldsymbol{\theta}}\right) \tag{38a}$$

$$\text{s.t. (37b) – (37c)}, \tag{38b}$$

where $\mathbf{u}_\mathrm{s} \sim \pi_{\boldsymbol{\theta}}[\,\cdot \,|\, \mathbf{x}\,]$ is a sample drawn from the stochastic policy. The cost function ϕ in (37) can be independent of $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, or, e.g., any metric in the functional space underlying the deterministic policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ and $\boldsymbol{\pi}^{\mathbb{S}}$. A similar construction can be done for ϕ in (38).

The following corollaries show that Propositions 1 and 2 hold in the context of (37) and (38) under some conditions.

Corollary 2. Proposition 1 holds for (37) with:

$$\nabla_{\mathbf{u}\boldsymbol{\theta}} \mathcal{L} = -\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}} + \nabla_{\mathbf{u}\boldsymbol{\theta}} \rho_{\boldsymbol{\theta}}(\mathbf{u}_0, \mathbf{x}), \tag{39a}$$

$$H = \mathbf{I} + \nabla^2_{\mathbf{u}_0} \rho_{\boldsymbol{\theta}}(\mathbf{u}_0, \mathbf{x}) + \nabla^2_{\mathbf{u}_0}\left(\boldsymbol{\mu}_{\mathbb{A}}^\top \mathbf{s}_{\mathbb{A}}\right). \tag{39b}$$

where

$$\rho_{\boldsymbol{\theta}}(\mathbf{u}_0, \mathbf{x}) = \min_{\boldsymbol{\pi}^{\mathbb{S}}} \quad \phi\left(\boldsymbol{\pi}^{\mathbb{S}}, \boldsymbol{\pi}_{\boldsymbol{\theta}}\right) \quad \text{s.t.} \quad \text{(37c)}, \tag{40a}$$

Proof. Problem (37) can be put in the form:

$$\mathbf{u}_0 = \arg\min_{\mathbf{u}_0} \ \frac{1}{2}\|\mathbf{u}_0 - \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{x})\|^2 + \rho_{\boldsymbol{\theta}}(\mathbf{u}_0, \mathbf{x}) \tag{41a}$$

$$\text{s.t.} \ \ \mathbf{s}(\mathbf{x}, \mathbf{u}_0) \le 0, \tag{41b}$$

One can then readily observe that Proposition 1 applies to (41), with (39). ■

Corollary 3. The results of Proposition 2 hold for (38) if function ϕ is independent of $\boldsymbol{\theta}$.

Proof. Problem (38) can be put in the form:

$$\mathbf{u}_0 = \arg\min_{\mathbf{u}_0} \ \frac{1}{2}\|\mathbf{u}_0 - \mathbf{u}_\mathrm{s}\|^2 + \rho_{\boldsymbol{\theta}}(\mathbf{u}_0, \mathbf{x}) \tag{42a}$$

$$\text{s.t.} \ \ \mathbf{s}(\mathbf{x}, \mathbf{u}_0) \le 0, \tag{42b}$$

where $\rho_{\boldsymbol{\theta}}$ is based on $\phi\left(\boldsymbol{\pi}^{\mathbb{S}}, \pi_{\boldsymbol{\theta}}\right)$. One can verify that Proposition 2 is independent of the choice of cost function in the projection as long as it is independent of $\boldsymbol{\theta}$, and holds as long as it satisfies LICQ/SOSC. As a result, if ϕ is independent of $\boldsymbol{\theta}$, Proposition 2 readily applies to (42). ■

If function ϕ depends on $\boldsymbol{\theta}$, more elaborate techniques must be used, see (Gros and Zanon, 2020).

6. SIMULATED EXAMPLE

In this section, we present a simple example illustrating Corollary 3. Let us consider the dynamic system:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \cos a & \sin a \\ -\sin a & \cos a \end{bmatrix} \mathbf{x}_k + \mathbf{u}_k + \mathbf{n}_k, \tag{43}$$

where $a = 20°$, $\mathbf{n}_k \in \mathbb{R}^2$ is truncated Normal centred of covariance $\Sigma_{\mathbf{n}} = 0.1\mathrm{I}$, and restricted to a ball of radius 0.1, i.e., $\mathbf{n}_k \in \mathcal{B}(0, 0.1)$. We consider a safety constraint:

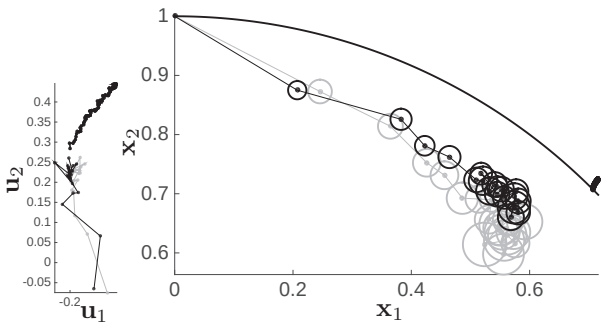

Fig. 3. Illustration of the input and state trajectories at the beginning (grey curves) and end (black curves) of the learning. The circles in the right graph display the state standard deviation. The markers show the evolution of the MPC references $\hat{\mathbf{u}}_{\text{ref}}$, $\hat{\mathbf{x}}_{\text{ref}}$.

$$\mathbf{h}\left(\mathbf{x}\right) = \mathbf{x}^\top \mathbf{x} - 1 \leq 0. \tag{44}$$

We will use the baseline cost:

$$L\left(\mathbf{x}, \mathbf{u}\right) = 10^{-2} \|\mathbf{x} - \mathbf{x}_{\text{ref}}\|^2 + \|\mathbf{u} - \mathbf{u}_{\text{ref}}\|^2. \tag{45}$$

The MPC will be based on the noise-free model

$$\bar{\mathbf{x}}_{k+1} = 1.1 \begin{bmatrix} \cos\hat{a} & \sin\hat{a} \\ -\sin\hat{a} & \cos\hat{a} \end{bmatrix} \bar{\mathbf{x}}_k + \mathbf{u}_k, \tag{46}$$

where $\hat{a} = 25°$, and the policy $\boldsymbol{\pi}^{\mathbb{S}}$ will be selected as:

$$\boldsymbol{\pi}^{\mathbb{S}}\left(\mathbf{x}, \mathbf{u}\right) = \mathbf{u} - K^{\mathbb{S}}\left(\mathbf{x} - \bar{\mathbf{x}}_k\right), \tag{47}$$

where $K^{\mathbb{S}}$ is the LQR corresponding to (46) for $Q, R = \text{I}$. We can represent the dispersion set as a ball, i.e.,

$$\mathbf{X}_k\left(\mathbf{x}, \mathbf{u}_0, \boldsymbol{\pi}^{\mathbb{S}}\right) = \mathcal{B}\left(\bar{\mathbf{x}}_k, r_k\right) \tag{48}$$

of radius $r_{k+1} = \|A\|_\infty r_k + \max_{\mathbf{n}\in\mathcal{B}(0,0.1)} \|\mathbf{n}\|$, and $r_0 = 0$. We then build the robust MPC scheme:

$$\mathbf{u}_{0,\ldots,N-1}(\mathbf{x}) = \underset{\mathbf{u}_{0,\ldots,N-1}}{\arg\min} \frac{1}{2}\|\mathbf{u}_0 - \mathbf{u}_{\text{s}}\|^2 + \sum_{k=1}^{N-1} \gamma^k L\left(\bar{\mathbf{x}}_k, \mathbf{u}_k\right) \tag{49a}$$

$$\text{s.t. (46)}, \quad \bar{\mathbf{x}}_0 = \mathbf{x}, \quad \mathbf{h}\left(\mathbf{x}_k\right) \leq 0 \tag{49b}$$

$$\forall \mathbf{x}_k \in \mathcal{B}\left(\bar{\mathbf{x}}_k, r_k\right), \quad \forall k. \tag{49c}$$

with $\gamma = 0.9$, and use $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\perp}\left(\mathbf{x}\right) = \mathbf{u}_0\left(\mathbf{x}\right)$. We consider the stochastic policy $\pi_{\boldsymbol{\theta}}$ delivering the samples $\mathbf{u}_{\text{s}}$ as Normal, centred at $\bar{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\left(\mathbf{x}\right)$, and of isotropic covariance $\sigma_\pi \text{I}$, where

$$\bar{\boldsymbol{\pi}}_{\boldsymbol{\theta}}\left(\mathbf{x}\right) = \hat{\mathbf{u}}_{\text{ref}} - K\left(\mathbf{x} - \hat{\mathbf{x}}_{\text{ref}}\right), \tag{50}$$

and the policy parameters are $\boldsymbol{\theta} = \{\hat{\mathbf{u}}_{\text{ref}}, \hat{\mathbf{x}}_{\text{ref}}, K\}$. A batch approach was used to compute the policy gradients, using (28), using 30 batches of duration 20, and LSTDV/LSTDQ techniques. The initial condition $\mathbf{x}_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$ was used. The MPC horizon is $N = 10$. A linear compatible advantage function approximator was used, built upon a quadratic value function approximation. Fig. 4 displays the evolution of the policy parameters through the learning, Fig. 5 shows the evaluation of the closed-loop performance, and Fig. 3 shows the evolution of the system trajectories through the learning process.

7. CONCLUSION

In this paper, we discussed the projection approach as a method to enforce the safety of a policy learned via RL. We showed that the approach is detrimental in the context of Q-learning, and that a direct minimization of the Q function under the safety constraints is arguably more suited. We showed that in the context for deterministic policies, the actor-critic method needs a simple correction in order for the policy gradient estimation to be unbiased. Similarly, in the context of stochastic policies, the actor-critic needs to be constructed in a very specific way in order for the policy gradient estimations to be unbiased. We showed that the results extend to the case of a projection performed via Robust MPC.

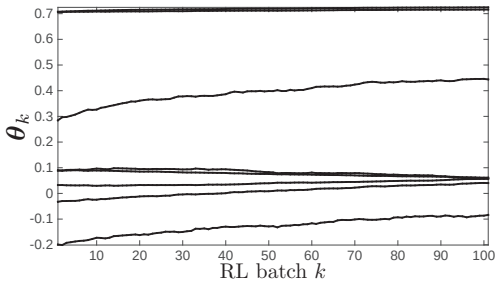

Fig. 4. Evaluation of the policy parameters $\boldsymbol{\theta}$ associated to the feedback matrix K in (50) over the learning

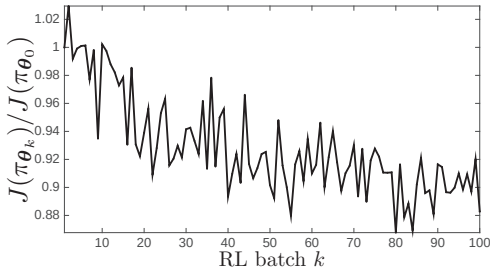

Fig. 5. Evaluation of the policy performance (1) over the learning, normed to 1.

REFERENCES AND NOTES

Bertsekas, D. (2007). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition.

Gros, S. and Zanon, M. (2020). Safe Reinforcement Learning Based on Robust MPC and Policy Gradient Methods. *IEEE Transactions on Automatic Control (submitted)*.

J. Garcia, J.F. (2013). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16, 1437–1480.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction. Second Edition*. MIT press Cambridge.

Sutton, R.S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, 1057–1063.

Wabersich, K., Hewing, L., Carron, A., and Zeilinger, M. (2019). Probabilistic model predictive safety certification for learning-based control. *arXiv:1906.10417v1, 25 Jun 2019*.

Zanon, M. and Gros (2019). Safe Reinforcement Learning Using Robust MPC. In *Transaction on Automatic Control, (submitted))*. Https://arxiv.org/abs/1906.04005.