Reinforcement learning for mixed-integer problems based on MPC

Reinforcement Learning for Mixed-Integer Problems Based on MPC

Sebastien Gros * **Mario Zanon** **

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

Abstract: Model Predictive Control has been recently proposed as policy approximation for Reinforcement Learning, offering a path towards safe and explainable Reinforcement Learning. This approach has been investigated for Q-learning and actor-critic methods, both in the context of nominal Economic MPC and Robust (N)MPC, showing very promising results. In that context, actor-critic methods seem to be the most reliable approach. Many applications include a mixture of continuous and integer inputs, for which the classical actor-critic methods need to be adapted. In this paper, we present a policy approximation based on mixed-integer MPC schemes, and propose a computationally inexpensive technique to generate exploration in the mixed-integer input space that ensures a satisfaction of the constraints. We then propose a simple compatible advantage function approximation for the proposed policy, that allows one to build the gradient of the mixed-integer MPC-based policy.

Keywords: Reinforcement Learning, Mixed-Integer Model Predictive Control, actor-critic methods, stochastic and deterministic policy gradient.

1. INTRODUCTION

Reinforcement Learning (RL) is a powerful tool for tackling stochastic processes without depending on a detailed model of the probability distributions underlying the state transitions. Indeed, most RL methods rely purely on observed data, and realizations of the stage cost assessing the system performance. RL methods seek to increase the closed-loop performance of the control policy deployed on the system as observations are collected. RL has drawn an increasingly large attention thanks to its accomplishments, such as, e.g., making it possible for robots to learn to walk or fly from experiments (Wang et al., 2012; Abbeel et al., 2007).

Most RL methods are based on learning the optimal control policy for the real system either directly, or indirectly. Indirect methods typically rely on learning a good approximation of the optimal action-value function underlying the system. The optimal policy is then indirectly obtained as the minimizer of the value-function approximation over the inputs. Direct RL methods, if based on the policy gradient, seek to adjust the parameters $\boldsymbol{\theta}$ of a given policy $\pi_{\boldsymbol{\theta}}$ such that it yields the best closed-loop performance when deployed on the real system. An attractive advantage of direct RL methods over indirect ones is that they are based on formal necessary conditions of optimality for the closed-loop performance of $\pi_{\boldsymbol{\theta}}$, and therefore guarantee - for a large enough data set - the (possibly local) assymptotic optimality of the parameters $\boldsymbol{\theta}$ (Sutton et al., 1999; Silver et al., 2014).

RL methods often rely on Deep Neural Networks (DNN) to carry the policy approximation $\pi_{\boldsymbol{\theta}}$. Unfortunately, control policies based on DNNs provide limited opportunities for formal verifications of the resulting policy, and for imposing hard constraints on the evolution of the state of the real system. The development of safe RL methods, which aims at tackling this issue, is currently an open field of research (J. Garcia, 2013). A novel approach towards providing formal safety certificates in the context of RL has been recently proposed in (Gros and Zanon, 2019, 2020; Zanon and Gros, 2019), where the policy approximation is based on robust Model Predictive Control (MPC) schemes rather than unstructured function approximators like DNNs. The validity of this choice is discussed in details in (Gros and Zanon, 2019). In (Gros and Zanon, 2020), methodologies to deploy direct RL techniques on MPC-based policy approximations are proposed. These methodologies are, however, restricted to continuous input spaces and therefore exclude integer decision variables, which are central in a number of applications.

In this paper, we propose an extension of the policy gradient techniques proposed in (Gros and Zanon, 2020) to mixed-integer problems. A mixed-integer MPC is used as a policy approximation, and a policy gradient method adjusts the MPC parameters for closed-loop performance. We detail how the actor-critic method can be deployed in this specific context. In particular, we propose an asymptotically exact hybrid stochastic-deterministic policy approach allowing for computing the policy gradient at a lower computational complexity than a full stochastic approach. We then propose a hybrid compatible advantage-function approximator tailored to our formulation. We finally detail how the mixed-integer MPC can be differentiated at a low computational cost, using principles from parametric Nonlinear Programming, in order to implement the actor-critic method. The proposed method is illustrated on a simple example, allowing for an unambiguous presentation of the results.

The paper is structured as follows. Section 2 provides background material on MDPs and RL. Section 3 presents the construction of a mixed-integer stochastic policy using a mixed-integer MPC scheme to support the policy approximation. Section 4 details an actor-critic method tailored to the proposed formulation, and how the policy gradient can be estimated. A compatible advantage function approximation is proposed. Section 5 details how the mixed-integer MPC scheme can be efficiently differentiated. Section 6 proposes an illustrative example, and Section 7 provides some discussions.

2. BACKGROUND

In the following, we will consider that the dynamics of the real system are described as a stochastic process on (possibly) continuous state-input spaces. We will furthermore consider (possibly) stochastic policies π, taking the form of probability densities:

$$\pi\left[\mathbf{a}\,|\,\mathbf{s}\right] : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}_+, \tag{1}$$

denoting the probability density of selecting a given input $\mathbf{a}$ when the system is in a given state $\mathbf{s}$. Deterministic policies delivering $\mathbf{a}$ as a function of $\mathbf{s}$ will be labelled as:

$$\boldsymbol{\pi}\left(\mathbf{s}\right) : \mathbb{R}^n \to \mathbb{R}^m. \tag{2}$$

Any deterministic policy can be viewed as a stochastic one, having a Dirac function as a probability density (or unit function for discrete inputs), i.e., $\pi\left[\mathbf{a}\,|\,\mathbf{s}\right] = \delta\left(\mathbf{a} - \boldsymbol{\pi}\left(\mathbf{s}\right)\right)$.

We consider a stage cost function $L(\mathbf{s},\mathbf{a}) \in \mathbb{R}$ and a discount factor $\gamma \in [0,1]$, the performance of a policy π is assessed via the total expected cost:

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k L(\mathbf{s}_k,\mathbf{a}_k) \,\middle|\, \mathbf{a}_k \sim \pi\left[\,.\,|\,\mathbf{s}_k\right]\right]. \tag{3}$$

The optimal policy associated to the state transition, the stage cost L and the discount factor γ is deterministic and given by:

$$\pi_\star = \arg\min_\pi\ J(\pi). \tag{4}$$

The value function $V_{\boldsymbol{\pi}}$, action-value function $Q_{\boldsymbol{\pi}}$ and advantage functions $A_{\boldsymbol{\pi}}$ associated to a given policy π are given by (Bertsekas, 1995; Bertsekas and Shreve, 1996; Bertsekas, 2007):

$$V_{\boldsymbol{\pi}}\left(\mathbf{s}\right) = \mathbb{E}\left[L(\mathbf{s},\mathbf{a}) + \gamma V_{\boldsymbol{\pi}}(\mathbf{s}_+)\,|\,\mathbf{s},\,\mathbf{a}\right], \tag{5a}$$

$$Q_{\boldsymbol{\pi}}\left(\mathbf{s},\mathbf{a}\right) = L(\mathbf{s},\mathbf{a}) + \gamma\mathbb{E}\left[V_{\boldsymbol{\pi}}(\mathbf{s}_+)\,|\,\mathbf{s},\,\mathbf{a}\right], \tag{5b}$$

$$A_{\boldsymbol{\pi}}\left(\mathbf{s},\mathbf{a}\right) = Q_{\boldsymbol{\pi}}\left(\mathbf{s},\mathbf{a}\right) - V_{\boldsymbol{\pi}}\left(\mathbf{s}\right), \tag{5c}$$

where the expected value in (5b) is taken over the state transition, and the one in (5a) is taken over the state transitions and (1).

2.1 Stochastic policy gradient

In most cases, the optimal policy $\pi_\star$ cannot be computed, either because the system is not exactly known or because solving (5) is too expensive. It is then useful to consider approximations $\pi_{\boldsymbol{\theta}}$ of the optimal policy, parametrized by $\boldsymbol{\theta}$. The optimal parameters $\boldsymbol{\theta}_\star$ are then given by:

$$\boldsymbol{\theta}_\star = \arg\min_{\boldsymbol{\theta}}\ J(\pi_{\boldsymbol{\theta}}). \tag{6}$$

The policy gradient $\nabla_{\boldsymbol{\theta}}\, J(\pi_{\boldsymbol{\theta}})$ associated to the stochastic policy $\pi_{\boldsymbol{\theta}}$ is then instrumental in finding $\boldsymbol{\theta}_\star$ by taking gradient steps in $\boldsymbol{\theta}$. The policy gradient can be obtained using various actor-critic methods (Sutton and Barto, 1998; Sutton et al., 1999). In this paper, we will use the actor-critic formulation:

$$\nabla_{\boldsymbol{\theta}}\, J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\, A_{\pi_{\boldsymbol{\theta}}}\right], \tag{7}$$

for stochastic policies, and the actor-critic formulation:

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

for deterministic policies.

The value functions $V_{\boldsymbol{\pi}}$, $Q_{\boldsymbol{\pi}}$ and $A_{\boldsymbol{\pi}}$ associated to a given policy π are typically evaluated via Temporal-Difference (TD) techniques (Sutton and Barto, 1998), and require that a certain amount of exploration is included in the deployment of the policy. For deterministic policies, the exploration can, e.g., be generated by including stochastic perturbations over the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, while stochastic policies generate exploration by construction. Note that is is fairly common in RL to define the stochastic policy $\pi_{\boldsymbol{\theta}}$ as an arbitrary density, e.g., the normal distribution, centered at a deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. We shall observe here that the deterministic policy gradient (8) is not suited as such for integer inputs, as the gradients $\nabla_{\boldsymbol{\theta}}\boldsymbol{\pi}_{\boldsymbol{\theta}}$, $\nabla_{\mathbf{a}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ do not exist on discrete input spaces. On continuous input spaces, the choice between the deterministic approach (8) or the stochastic approach (7) is typically motivated by computational aspects.

3. MIXED-INTEGER OPTIMIZATION-BASED POLICY

In this paper, we will consider parametrized deterministic policies $\boldsymbol{\pi}_{\boldsymbol{\theta}} \approx \boldsymbol{\pi}_\star$ based on parametric optimization problems. In particular, we will focus on optimization problems resulting from a nominal mixed-integer MPC formulation. The results proposed in this paper extend to robust MPC - enabling the construction of safe Reinforcement Learning methods - but this case is omitted in this paper for the sake of brevity.

3.1 Policy approximation based on mixed-integer MPC

The mixed-integer MPC scheme reads as:

$$\mathbf{u}^\star\left(\mathbf{s},\boldsymbol{\theta}\right), \mathbf{i}^\star\left(\mathbf{s},\boldsymbol{\theta}\right) =$$

$$\arg\min_{\mathbf{u},\mathbf{i}} \quad T(\mathbf{x}_N,\boldsymbol{\theta}) + \sum_{k=0}^{N-1} \ell(\mathbf{x}_k,\mathbf{u}_k,\mathbf{i}_k,\boldsymbol{\theta}) \tag{9a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{F}\left(\mathbf{x}_k,\mathbf{u}_k,\mathbf{i}_k,\boldsymbol{\theta}\right),\ \mathbf{x}_0 = \mathbf{s}, \tag{9b}$$

$$\mathbf{h}_k\left(\mathbf{x}_k,\mathbf{u}_k,\mathbf{i}_k,\boldsymbol{\theta}\right) \le 0, \quad k = 0,...,N-1, \tag{9c}$$

$$\mathbf{h}_N\left(\mathbf{x}_N,\boldsymbol{\theta}\right) \le 0, \tag{9d}$$

$$\mathbf{i}_k \in \{0,1\}^{m_{\mathrm{i}}}, \tag{9e}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ are the predicted system trajectories, $\mathbf{u}_k \in \mathbb{R}^{m_{\mathrm{c}}}$ the planned continuous inputs and $\mathbf{i}_k \in \{0,1\}^{m_{\mathrm{i}}}$ the planned integer inputs. Without loss of generality, we consider binary integer inputs. Functions ℓ, T are the stage and terminal costs. Functions $\mathbf{h}_{0,\dots,N-1}$ are the stage constraints and function $\mathbf{h}_N$ is the terminal constraint.

For a given state $\mathbf{s}$ and parameters $\boldsymbol{\theta}$, the MPC scheme (9) delivers the continuous and integer input profiles

$$\mathbf{u}^\star\left(\mathbf{s},\boldsymbol{\theta}\right) = \left\{\mathbf{u}_0^\star\left(\mathbf{s},\boldsymbol{\theta}\right),\dots,\mathbf{u}_{N-1}^\star\left(\mathbf{s},\boldsymbol{\theta}\right)\right\}, \tag{10a}$$

$$\mathbf{i}^\star\left(\mathbf{s},\boldsymbol{\theta}\right) = \left\{\mathbf{i}_0^\star\left(\mathbf{s},\boldsymbol{\theta}\right),\dots,\mathbf{i}_{N-1}^\star\left(\mathbf{s},\boldsymbol{\theta}\right)\right\}, \tag{10b}$$

with $\mathbf{u}_k^\star(\mathbf{s},\boldsymbol{\theta}) \in \mathbb{R}^{m_\mathrm{c}}$ and $\mathbf{i}_k^\star(\mathbf{s},\boldsymbol{\theta}) \in \{0,1\}^{m_\mathrm{i}}$. The MPC scheme (9) generates a parametrized deterministic policy

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) = \left\{\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{c}}(\mathbf{s}), \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{i}}(\mathbf{s})\right\}, \tag{11}$$

where

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{c}}(\mathbf{s}) = \mathbf{u}_0^\star(\mathbf{s},\boldsymbol{\theta}) \in \mathbb{R}^{m_\mathrm{c}}, \tag{12a}$$

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{i}}(\mathbf{s}) = \mathbf{i}_0^\star(\mathbf{s},\boldsymbol{\theta}) \in \{0,1\}^{m_\mathrm{i}}, \tag{12b}$$

are the first elements of the continuous and integer input sequences generated by (9). In the following, it will be useful to consider the MPC scheme (9) as a generic parametric mixed-integer NLP:

$$\begin{aligned}\mathbf{u}^\star(\mathbf{s},\boldsymbol{\theta}), \mathbf{i}^\star(\mathbf{s},\boldsymbol{\theta}) = \arg\min_{\mathbf{u},\mathbf{i}} \quad & \Phi(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) && \text{(13a)}\\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{i},\mathbf{s},\boldsymbol{\theta}) = 0, && \text{(13b)}\\ & \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) \leq 0, && \text{(13c)}\\ & \mathbf{i} \in \{0,1\}^{m_\mathrm{i}\times N-1}, && \text{(13d)}\end{aligned}$$

where function Φ gathers the stage and terminal cost functions from (9a), function $\mathbf{f}$ gathers the dynamic constraints and initial conditions (9b), and function $\mathbf{h}$ gathers the stage and terminal constraints (9c)-(9d).

4. ACTOR-CRITIC METHOD

In order to build actor-critic methods for (11), exploration is required (Sutton and Barto, 1998). When the input space is constrained and mixed-integer, the exploration becomes non-trivial to setup, as 1. it must retain the feasibility of the hard constraints (9c)-(9d) and 2. simple input disturbances are not possible for the integer part since they are locked on an integer grid. To address this issue, we will adopt a stochastic policy approach, well suited for the integer part, and consider its asymptotically equivalent deterministic counterpart on the continuous input space, well suited for computational efficiency.

4.1 MPC-based exploration

In order to generate exploration, we will build a stochastic policy (1) based on the deterministic policy (11) where $\mathbf{a}$ will gather the continuous inputs $\mathbf{a}^{\mathrm{c}}$ and integer inputs $\mathbf{a}^{\mathrm{i}}$ actually applied to the real system, i.e., $\mathbf{a} = \left\{\mathbf{a}^{\mathrm{c}}, \mathbf{a}^{\mathrm{i}}\right\}$. We will build (1) such that it generates exploration that is respecting the constraints (9c)-(9d) with unitary probability. We propose to build (1) such that it becomes naturally separable between the integer and continuous part in the policy gradient computation. To that end, we consider a softmax approach to handle the integer part of the problem. More specifically, we consider the parametric mixed-integer NLP:

$$\begin{aligned}\Phi^{\mathrm{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}) = \min_{\mathbf{u},\mathbf{i}} \quad & \Phi(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) && \text{(14a)}\\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{i},\mathbf{s},\boldsymbol{\theta}) = 0, && \text{(14b)}\\ & \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) \leq 0, && \text{(14c)}\\ & \mathbf{i}_0 = \mathbf{a}^{\mathrm{i}}, && \text{(14d)}\\ & \mathbf{i}_{1,\ldots,N-1} \in \{0,1\}^{m_\mathrm{i}}, && \text{(14e)}\end{aligned}$$

derived from (13), where the first integer input is assigned to $\mathbf{a}^{\mathrm{i}}$ via constraint (14d). We will consider that $\Phi^{\mathrm{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}})$ takes infinite value when the selected integer input $\mathbf{a}^{\mathrm{i}}$ is infeasible. Let us label $\mathbb{I}(\mathbf{s},\boldsymbol{\theta})$ the feasible set of $\mathbf{a}^{\mathrm{i}}$ for a given state $\mathbf{s}$ and MPC parameter $\boldsymbol{\theta}$, and $\tilde{\mathbf{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}})$ the integer profile solution of (14). By construction $\tilde{\mathbf{i}}_0(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}) = \mathbf{a}^{\mathrm{i}}$ when $\mathbf{a}^{\mathrm{i}} \in \mathbb{I}(\mathbf{s},\boldsymbol{\theta})$. We then define the softmax stochastic integer policy distribution using

$$\pi_{\boldsymbol{\theta}}^{\mathrm{i}}\left[\,\mathbf{a}^{\mathrm{i}}\,|\,\mathbf{s}\,\right] \propto e^{-\sigma_\mathrm{i}^{-1}\Phi_\mathrm{i}^\star(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}})} \in \mathbb{R}_+, \tag{15}$$

where $\sigma_\mathrm{i} > 0$ is a parameter adjusting the variance of $\pi_{\boldsymbol{\theta}}^{\mathrm{i}}$. In order to build the continuous part of the policy, we will consider the continuous part $\mathbf{a}^{\mathrm{c}}$ of the stochastic policy as conditioned on $\tilde{\mathbf{i}}$, and taking the form of a probability density:

$$\pi_{\boldsymbol{\theta}}^{\mathrm{c}}\left[\,\mathbf{a}^{\mathrm{c}} \mid \tilde{\mathbf{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}),\, \mathbf{s}\,\right] \in \mathbb{R}_+, \tag{16}$$

which will be constructed from the parametric NLP:

$$\begin{aligned}\tilde{\mathbf{u}}(\mathbf{s},\boldsymbol{\theta},\mathbf{i},\mathbf{d}) = \arg\min_{\mathbf{u}} \quad & \Phi(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) + \mathbf{d}^\top\mathbf{u}_0 && \text{(17a)}\\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{i},\mathbf{s},\boldsymbol{\theta}) = 0, && \text{(17b)}\\ & \mathbf{h}(\mathbf{x},\mathbf{u},\mathbf{i},\boldsymbol{\theta}) \leq 0, && \text{(17c)}\end{aligned}$$

derived from (13), but where the integer input profile is entirely assigned, and where $\mathbf{d} \in \mathrm{R}^{m_\mathrm{c}}$ is a random vector chosen as $\mathbf{d} \sim \mathcal{N}(0, \sigma_\mathrm{c}\Sigma)$. The random variable $\mathbf{a}_\mathrm{c}$ in (16) will then be selected as:

$$\mathbf{a}^{\mathrm{c}} = \tilde{\mathbf{u}}_0(\mathbf{s},\boldsymbol{\theta},\tilde{\mathbf{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}),\mathbf{d}). \tag{18}$$

As previously observed in (Gros and Zanon, 2020), while $\pi_{\boldsymbol{\theta}}^{\mathrm{c}}$ is easy to sample, it is in general difficult to evaluate.

Because $\mathbf{a}^{\mathrm{c}}$ is conditioned on $\tilde{\mathbf{i}}$ and, therefore, $\mathbf{a}^{\mathrm{i}}$, the Kolmogorov definition of conditional probabilities entails that the overall stochastic policy (1) reads as the distribution:

$$\pi_{\boldsymbol{\theta}}[\,\mathbf{a}\,|\,\mathbf{s}\,] = \pi_{\boldsymbol{\theta}}^{\mathrm{c}}\left[\,\mathbf{a}^{\mathrm{c}} \,\middle|\, \tilde{\mathbf{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}),\, \mathbf{s}\,\right] \pi_{\boldsymbol{\theta}}^{\mathrm{i}}\left[\,\mathbf{a}^{\mathrm{i}}\,|\,\mathbf{s}\,\right]. \tag{19}$$

We establish next a straightforward but useful result concerning the stochastic policy (19).

Lemma 1. The stochastic policy (19) generates input samples $\mathbf{a}$ that are feasible for the MPC scheme (9).

Proof. Because $\Phi^{\mathrm{i}}(\mathbf{s},\boldsymbol{\theta},\mathbf{a}^{\mathrm{i}}) = +\infty$ when $\mathbf{a}^{\mathrm{i}} \notin \mathbb{I}(\mathbf{s},\boldsymbol{\theta})$, policy (15) selects feasible integer inputs $\mathbf{a}^{\mathrm{i}}$ with probability 1. Furthermore, NLP (17) is feasible for all $\mathbf{a}^{\mathrm{i}} \in \mathbb{I}(\mathbf{s},\boldsymbol{\theta})$ and all $\mathbf{d}$, such that its solution satisfies constraints (13b)-(13c). As a result, the samples $\mathbf{a}^{\mathrm{i}}, \mathbf{a}^{\mathrm{c}}$ generated from (19) are guaranteed to be feasible. □

The policy gradient associated to (19) can be computed using (7). Unfortunately, it has been observed that this approach is computationally expensive for continuous input spaces (Gros and Zanon, 2020) when the policy is restricted by non-trivial constraints. Hence, we now turn to detailing how the policy gradient associated to policy (19) can be efficiently computed.

4.2 Policy gradient

Using policy (19), the stochastic policy gradient is separable between the continuous and integer part and reads as:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) &= \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] && \text{(20)}\\ &= \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}^{\mathrm{c}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] + \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}^{\mathrm{i}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right],\end{aligned}$$

where $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ is the advantage function associated to the stochastic policy (19). Using (15), we then observe that the score function associated to the integer part of the policy is simply given by:

$$\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}}[\,\mathbf{a}^{\mathrm{i}} \,|\, \mathbf{s}\,] = -\frac{1}{\sigma_{\mathrm{i}}} \nabla_{\boldsymbol{\theta}} \Phi^{\star}_{\mathrm{i}}(\mathbf{s}, \boldsymbol{\theta}, \mathbf{a}^{\mathrm{i}}) + \frac{1}{\sigma_{\mathrm{i}}} \sum_{\mathbf{i}_0 \in \mathbb{I}(\mathbf{s}, \boldsymbol{\theta})} \pi^{\mathrm{i}}_{\boldsymbol{\theta}}\,[\,\mathbf{i}_0 \,|\, \mathbf{s}\,]\, \nabla_{\boldsymbol{\theta}} \Phi^{\star}_{\mathrm{i}}(\mathbf{s}, \boldsymbol{\theta}, \mathbf{i}_0). \tag{21}$$

The computation of the policy gradient associated to the continuous part of the stochastic policy ought to be treated differently. Indeed, it has been observed in (Gros and Zanon, 2020) that deterministic policy gradient methods are computationally more effective than stochastic ones for policy approximations on problems having continuous input and state spaces. Defining the deterministic policy for the continuous inputs $\mathbf{a}^{\mathrm{c}}$ as

$$\boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{i}) = \tilde{\mathbf{u}}_0(\mathbf{s}, \boldsymbol{\theta}, \mathbf{i}, 0), \tag{22}$$

where $\tilde{\mathbf{u}}_0$ is the first element of the solution of (17), we consider the approximation (Silver et al., 2014)

$$\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{c}}_{\boldsymbol{\theta}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] \approx \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}} \nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right], \tag{23}$$

which is asymptotically exact for $\sigma_{\mathrm{c}} \to 0$ under some technical but fairly unrestrictive assumptions. We can then use the asymptotically exact hybrid policy gradient

$$\widehat{\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})} = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}} \nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] + \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right], \tag{24}$$

as a computationally effective policy gradient evaluation. The stochastic policy (16) is then deployed on the system and generates exploration, while the deterministic policy (22) is used to compute the policy gradient (24). We propose next a compatible advantage function approximator for (24), offering a systematic approximation of the advantage function $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$.

4.3 Compatible advantage function approximation

We note that the advantage function approximation

$$\hat{A}_{\pi_{\boldsymbol{\theta}}} = \mathbf{w}^{\top} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} = \mathbf{w}^{\top} \nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} + \mathbf{w}^{\top} \nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{c}}_{\boldsymbol{\theta}}, \tag{25}$$

is compatible by construction (Silver et al., 2014) for the stochastic policy gradient (20), in the sense that

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] \tag{26}$$

holds if $\mathbf{w}$ is the solution of the Least-Squares problem

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)^2\right]. \tag{27}$$

Similarly, we seek a compatible advantage function approximation for the hybrid policy gradient (24). We propose the hybrid advantage function approximation, inspired from (Gros and Zanon, 2020):

$$\hat{A}_{\pi_{\boldsymbol{\theta}}} = \mathbf{w}^{\top} \nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} + \mathbf{w}^{\top} \frac{1}{\sigma_{\mathrm{c}}} \nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M\,(\mathbf{e} - \mathbf{c}), \tag{28}$$

where we label $\mathbf{e} = \mathbf{a}^{\mathrm{c}} - \boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}}$ the exploration performed on the continuous part of the input space $\mathbb{R}^{m_{\mathrm{c}}}$, and $M \in \mathbb{R}^{m_{\mathrm{c}} \times m_{\mathrm{c}}}$ is symmetric and $\mathbf{c} \in \mathbb{R}^{m_{\mathrm{c}}}$. We will show in the following proposition that for M and $\mathbf{c}$ adequately chosen, the advantage function approximation (28) is compatible with the policy gradient (24).

Proposition 1. The hybrid function approximation (28) is asymptotically compatible, i.e.,

$$\lim_{\sigma_{\mathrm{c}} \to 0} \widehat{\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}})} = \lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}} \nabla_{\mathbf{a}^{\mathrm{c}}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] + \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right] \tag{29}$$

holds for $\mathbf{w}$ solution of (27) and for M, $\mathbf{c}$ chosen according to (Gros and Zanon, 2020):

$$\mathbf{c} = \frac{\sigma_{\mathrm{c}}}{2} \sum_{i,j=1}^{n_{\mathbf{a}}} \frac{\partial^2 \tilde{\mathbf{u}}_0}{\partial \mathbf{d}_i \mathbf{d}_j} \Sigma_{ij}, \quad M = \left(\frac{\partial \tilde{\mathbf{u}}_0}{\partial \mathbf{d}} \Sigma \frac{\partial \tilde{\mathbf{u}}_0}{\partial \mathbf{d}}^{\top}\right), \tag{30}$$

evaluated at the solution of (17) for $\mathbf{d} = 0$, where (17) satisfies the regularity assumptions of (Gros and Zanon, 2020, Proposition 1). These assumptions are technical but fairly unrestrictive, see (Gros and Zanon, 2020) for a complete discussion.

The proof delivered below is a sketch that follows the lines of the proof of Proposition 1 in Gros and Zanon (2020).

Proof. We observe that the solution $\mathbf{w}$ of (27) using (28) is given by:

$$\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\left(\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} + \frac{1}{\sigma_{\mathrm{c}}} \nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M\,(\mathbf{e} - \mathbf{c})\right)\left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] = 0. \tag{31}$$

Using a Taylor expansion of $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ at $\mathbf{e} = 0$, as proposed in (Gros and Zanon, 2020, Proposition 1), we observe that (31) becomes:

$$\begin{aligned} &\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} \left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] + \frac{1}{\sigma_{\mathrm{c}}} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M\,(\mathbf{e} - \mathbf{c})\,\xi\right] \\ &+ \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M \frac{\mathbf{e} - \mathbf{c}}{\sigma_{\mathrm{c}}} \left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] + \\ &+ \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M \frac{(\mathbf{e} - \mathbf{c})\,\mathbf{e}^{\top}}{\sigma_{\mathrm{c}}} \left(\nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \nabla_{\mathbf{a}^{\mathrm{c}}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] = 0, \end{aligned} \tag{32}$$

where ξ is the second-order remainder of the Taylor expansion of $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$. Unlike (31), all terms in (32) are evaluated at $\mathbf{s}$, $\mathbf{a}^{\mathrm{c}} = \boldsymbol{\pi}^{\mathrm{c}}(\mathbf{s})$. Following a similar argumentation as in (Gros and Zanon, 2020, Proposition 1), we obtain

$$\begin{aligned} &\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\frac{1}{\sigma_{\mathrm{c}}} \nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M\,(\mathbf{e} - \mathbf{c})\,\mathbf{e}^{\top} \left(\nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \nabla_{\mathbf{a}^{\mathrm{c}}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] \\ &\qquad = \lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} \left(\nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \nabla_{\mathbf{a}^{\mathrm{c}}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right], \end{aligned} \tag{33a}$$

$$\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\frac{1}{\sigma_{\mathrm{c}}} \nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M\,(\mathbf{e} - \mathbf{c})\,\xi\right] = 0, \tag{33b}$$

$$\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} M \frac{\mathbf{e} - \mathbf{c}}{\sigma_{\mathrm{c}}} \left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)_{\mathbf{e}=0}\right] = 0. \tag{33c}$$

Equality (33b) holds from the Delta method, while equalities (33a), (33c) hold because

$$\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\frac{1}{\sigma_{\mathrm{c}}} M\,(\mathbf{e} - \mathbf{c})\,\mathbf{e}^{\top}\right] = I, \tag{34}$$

$$\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[M \frac{\mathbf{e} - \mathbf{c}}{\sigma_{\mathrm{c}}}\right] = 0, \tag{35}$$

result from (30), see (Gros and Zanon, 2020). Hence

$$\begin{aligned} &\lim_{\sigma_{\mathrm{c}} \to 0} \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}} \left(A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] \\ &\qquad + \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \pi^{\mathrm{c}}_{\boldsymbol{\theta}} \left(\nabla_{\mathbf{a}^{\mathrm{c}}} A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} - \nabla_{\mathbf{a}^{\mathrm{c}}} \hat{A}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\right)\right] = 0. \end{aligned} \tag{36}$$

Using (24), (29) holds from (36). □

5. NLP SENSITIVITIES

In order to deploy the policy gradient techniques described above, one needs to compute the sensitivities $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}^{\mathrm{c}}_{\boldsymbol{\theta}}$ and $\nabla_{\boldsymbol{\theta}} \log \pi^{\mathrm{i}}_{\boldsymbol{\theta}}$. Computing the score function (21) requires

computing the sensitivity of the cost function $\Phi_{\mathrm{i}}^{\star}$ of the NLP (14). This sensitivity exists almost everywhere and is given by:

$$\nabla_{\boldsymbol{\theta}} \Phi_{\mathrm{i}}^{\star}(\mathbf{s}, \boldsymbol{\theta}, \mathbf{a}^{\mathrm{i}}) = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\theta}, \mathbf{a}^{\mathrm{i}}, \mathbf{d}), \tag{37}$$

where $\mathbf{y}$ is the primal solution of the NLP (14), gathering the continuous inputs and states of the NLP, and $\boldsymbol{\lambda}, \boldsymbol{\mu}$ the dual variables associated to constraints (13b)-(13c), respectively, and $\mathcal{L} = \Phi + \mathbf{d}^{\top}\mathbf{u}_0 + \boldsymbol{\lambda}^{\top}\mathbf{f} + \boldsymbol{\mu}^{\top}\mathbf{h}$ is the Lagrange function associated to (14). The computation of $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{c}}$ is more involved. Consider:

$$\mathbf{r} = \begin{bmatrix} \nabla_{\mathbf{y}} \mathcal{L}(\mathbf{z}, \mathbf{s}, \boldsymbol{\theta}, \mathbf{a}^{\mathrm{i}}, \mathbf{d}) \\ \mathbf{f}\left(\mathbf{w}, \mathbf{s}, \boldsymbol{\theta}\right) \\ \mathrm{diag}(\boldsymbol{\mu})\mathbf{h}\left(\mathbf{w}, \boldsymbol{\theta}\right) + \tau \end{bmatrix} = 0, \tag{38}$$

i.e., the primal-dual interior-point KKT conditions associated to (14) for a barrier parameter $\tau > 0$, and $\mathbf{z}$ gathering the primal-dual variables of the NLP (14), i.e., $\mathbf{z} = \{\mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}\}$. Then, if the solution of the NLP (14) satisfies LICQ and SOSC (Nocedal and Wright, 2006), the sensitivity of the solution of the NLP (14) exists almost everywhere and can be computed via the Implicit Function Theorem, providing

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}} = -\frac{\partial \mathbf{r}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{r}}{\partial \boldsymbol{\theta}}, \tag{39}$$

see (Büskens and Maurer, 2001). Using (22), the sensitivity $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{c}}$ then read as

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{c}} = \nabla_{\boldsymbol{\theta}} \tilde{\mathbf{u}}_0, \tag{40}$$

where $\nabla_{\boldsymbol{\theta}} \tilde{\mathbf{u}}_0$ is extracted from $\frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}$.

6. SIMULATED EXAMPLE

For the sake of brevity and in order to present results that are easy to interpret and verify, we propose to use a very low dimensional example, allowing us to bypass the evaluation of the action-value function via Temporal-Difference techniques, and isolate the discussions of this paper from questions regarding TD methods. We consider the linear, scalar dynamics:

$$s_{k+1} = s_k + a_k^{\mathrm{c}}\, i_k + n_k \tag{41}$$

where $s_k, a_k^{\mathrm{c}} \in \mathbb{R}$, $i_k \in \{0, 1\}$ and n_k is uniformly distributed in $[0,\ 0.05]$. We consider the baseline stage cost:

$$\begin{aligned} L\left(s, \mathbf{a}\right) =& \frac{1}{2}(s - s_{\mathrm{ref}})^2 + \frac{1}{2}(a^{\mathrm{c}} - a_{\mathrm{ref}}^{\mathrm{c}})^2 + w\, i \\ &+ c\, \max\left(|s| - 0.2, 0\right), \end{aligned} \tag{42}$$

as the reference performance, where w, $c \in \mathbb{R}_+$ are scalar weight and s_{ref}, a_{ref} are references for the state and continuous input. The MPC model is deterministic, given by:

$$x_{k+1} = x_k + u_k\, i_k + b \tag{43}$$

where $b \in \mathbb{R}$ is constant, but subject to adaptation via RL. The baseline cost imposes a high penalty for $s \notin [-0.2, 0.2]$, and constitutes an exact relaxation of the constraint $-0.2 \leq s \leq 0.2$, see (Gros and Zanon, 2019). The MPC stage cost ℓ has the form (42). The MPC parameters s_{ref}, $a_{\mathrm{ref}}^{\mathrm{c}}$, c and b are subject to adaptation via RL.

The policy gradient (29) was implemented, where the advantage function estimation was computed from (27), using the approximator $\hat{A}_{\pi_{\boldsymbol{\theta}}}$ from (28). The true advantage function $A_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ was evaluated via classic policy evaluation (Sutton and Barto, 1998) in order to deliver unambiguous results. On more complex examples (27) would be evaluated via Temporal-Difference techniques. The evaluations of (29) and (27) were performed in a batch fashion, using 30 batches of 50 time steps each, all starting from the deterministic initial condition $\mathbf{s}_0 = 0$. The MPC scheme had a horizon of $N = 10$ time samples, and a terminal cost based on the Riccati matrix of the control problem with $i = 1$. A discount factor of $\gamma = 0.95$ was adopted. The step-size selected for adapting the parameters from the policy gradient was $\alpha = 2 \cdot 10^{-3}$. The exploration parameters were chosen as $\sigma_{\mathrm{i}} = 2 \cdot 10^{-2}$, $\sigma_{\mathrm{c}} = 10^{-2}$ and $\Sigma = I$.

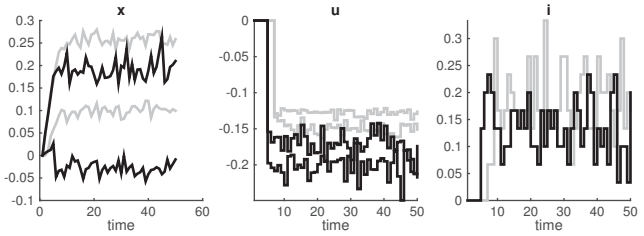

Fig. 1. Closed-loop trajectories before (light grey) and after (black) the learning. The left graph shows the extreme values of the state trajectories, the middle graph shows the extreme values of the continuous input a_k^{c} when $i_k = 1$, and the right graph shows the proportion of $i_k = 1$.

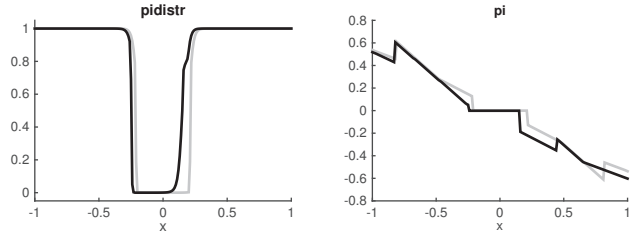

Fig. 2. Policy before (light grey) and after (black) the learning. The left graph shows the Softmax policy (15) as a function of the state s, giving the probability of selecting $i = 1$ for a given state s. The right graph shows the MPC policy (without the stochastic choice of integer variable).

The parameters $s_{\mathrm{ref}} = a_{\mathrm{ref}}^{\mathrm{c}} = 0$, $w = 0.2$, $c = 1$ were adopted for the baseline cost. The MPC scheme parameters were initialized using the same values, and using $b = 0$. Fig. 1 reports the trajectories of the system at the beginning and end of the learning process, showing how performance is gained by bringing the state trajectories in the interval $[-0.2, 0.2]$. Fig. 2 reports the policy for the continuous and integer inputs, showing how RL reshapes the MPC policy for a better closed-loop performance. Fig. 3 reports the estimated policy gradients via the compatible approximation (29) and directly via (24), showing a match predicted by Prop. 1. Fig. 4 reports the closed-loop performance of the MPC controller, calculated from $J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right) = V_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\mathbf{s}_0)$, and shows the performance gain obtained via the learning. Fig. 5 shows the MPC parameter evolution through the learning process.

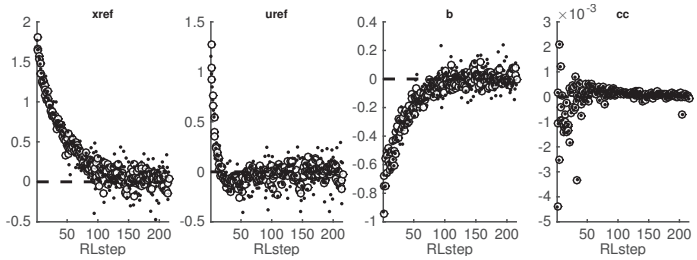

Fig. 3. Policy gradients throughout the learning process (iterations of the RL method). The dots display the policy gradient as obtained from (29), while the circles display the policy gradients obtained from (24).

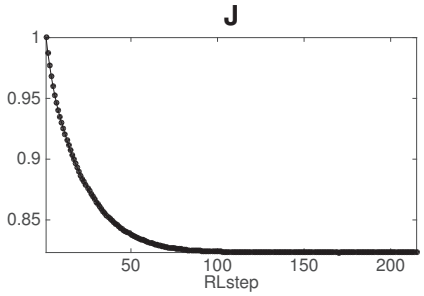

Fig. 4. Evolution of the closed-loop relative performance throughout the learning process. A reduction of the cost of over 15% is achieved here from $\boldsymbol{\theta}_0$.

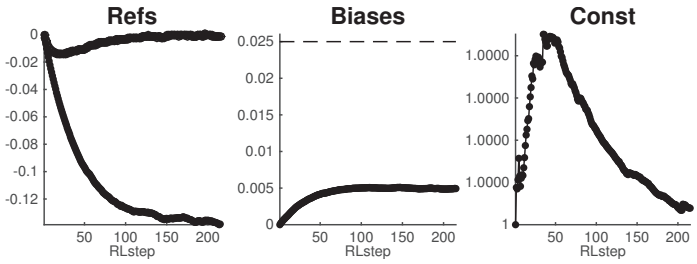

Fig. 5. Evolution of the MPC parameters throughout the learning process. The references are adjusted so that the system trajectories are better contained in the interval $[-0.2, 0.2]$. The model bias b does not match the value that a classic Prediction Error Method would deliver ($b = \mathbb{E}[n_k] = 0.025$, dashed line), while the cost associated to constraints is left unchanged.

7. DISCUSSION & CONCLUSION

This paper proposed an actor-critic approach to compute the policy gradient associated to policy approximations based on mixed-integer MPC schemes. The methodology is generic and applicable to linear, nonlinear and robust approaches. The paper proposes a hybrid stochastic-deterministic policy approach to generate the exploration and evaluate the policy gradient, avoiding the heavy computational expenses associated to using a stochastic policy approach on problems having continuous inputs and state constraints. A simple, compatible advantage function approximation is then proposed, tailored to our formulation and to MPC-based policy approximations. Some implementation details are provided, and the methods are illustrated on a simple example, providing a clear picture of how the proposed method is performing.

Future work will consider extensions to reduce the noise in the policy gradient estimation resulting from the choice of advantage function approximation, and will investigate techniques to integrate the stochastic policy and sensitivity computations with the branch-and-bound techniques used to solve the mixed-integer MPC problem. Future work will also investigate the potential of using the approaches detailed here to offer computationally less expensive approaches to solve the mixed-integer problem.

REFERENCES AND NOTES

Abbeel, P., Coates, A., Quigley, M., and Ng, A.Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, 2007. MIT Press.

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

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, Belmont, MA.

Bertsekas, D. and Shreve, S. (1996). *Stochastic Optimal Control: The Discrete Time Case*. Athena Scientific, Belmont, MA.

Büskens, C. and Maurer, H. (2001). *Online Optimization of Large Scale Systems*, chapter Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems, 3–16. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gros, S. and Zanon, M. (2019). Data-Driven Economic NMPC using Reinforcement Learning. *IEEE Transactions on Automatic Control*. (in press).

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.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition.

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*, ICML'14, I–387–I–395.

Sutton, R.S. and Barto, A.G. (1998). *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition.

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. MIT Press, Cambridge, MA, USA.

Wang, S., Chaovalitwongse, W., and Babuska, R. (2012). Machine learning algorithms in bipedal robot control. *Trans. Sys. Man Cyber Part C*, 42(5), 728–743.

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