Safe Reinforcement Learning Using Robust MPC

Mario Zanon, *Member, IEEE*, and Sebastien Gros

***Abstract*—Reinforcement learning (RL) has recently impressed the world with stunning results in various applications. While the potential of RL is now well established, many critical aspects still need to be tackled, including safety and stability issues. These issues, while secondary for the RL community, are central to the control community that has been widely investigating them. Model predictive control (MPC) is one of the most successful control techniques because, among others, of its ability to provide such guarantees even for uncertain constrained systems. Since MPC is an optimization-based technique, optimality has also often been claimed. Unfortunately, the performance of MPC is highly dependent on the accuracy of the model used for predictions. In this article, we propose to combine RL and MPC in order to exploit the advantages of both, and therefore, obtain a controller that is optimal and safe. We illustrate the results with two numerical examples in simulations.**

***Index Terms*—Reinforcement learning (RL), robust model predictive control (MPC), safe policies.**

I. Introduction

REINFORCEMENT learning (RL) is a technique for solving problems involving Markov decision processes (MDP) [1]. In RL, rather than modeled state transition probabilities, samples and observed costs (or rewards) are used. RL algorithms enabled computers beating Chess and Go masters [2], and robots learning to walk or fly without supervision [3], [4].

For each state $\mathbf{s}$, the optimal action $\mathbf{a}$ is computed as the the optimal feedback policy $\pi(\mathbf{s})$ for the real system either directly (policy search methods) [5], [6] or indirectly (SARSA, Q-learning) [7]. In the latter, the optimal policy $\pi(\mathbf{s})$ is indirectly obtained as the minimizer of the so-called action-value function $Q(\mathbf{s}, \mathbf{a})$ over the action or input $\mathbf{a}$. In both cases, either π or Q are typically approximated by a function approximator: Deep neural networks (DNNs) are very commonly used for that purpose in recent applications.

Manuscript received June 19, 2020; accepted September 5, 2020. Date of publication September 15, 2020; date of current version July 28, 2021. Recommended by Associate Editor E. C. Kerrigan. *(Corresponding author: Mario Zanon.)*

Mario Zanon is with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: mario.zanon@imtlucca.it).

Sebastien Gros is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology 7491 Trondheim, Norway (e-mail: grosse@chalmers.se).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2020.3024161

While RL has demonstrated in practice a huge potential, properties that are typically expected from a controller, such as, e.g., some form of stability and safety, are hard to guarantee, especially when relying on a DNN as a function approximator. In general, any safety-enforcing approach requires either to collect data from the real system thus incurring catastrophic events, to use a model of the system to generate a sufficient amount of stochastic simulations, or to model the uncertainty underlying the system, as we will detail later. Some approaches have been developed in order to guarantee some form of safety (see, e.g., the excellent survey in [8] and references therein). Most approaches, however, do not strictly guarantee that a given set of constraints is never violated, but rather that violations are rare events. Some approaches propose to project the action resulting from a DNN onto a safe set: in [9], each constraint is approximated as a rectified linear unit or an additional cost, and in [10], a quadratic program (QP) is used. In both approaches, the nominal prediction is used and uncertainty is neglected in the predictions, similarly to the approach of [11]. Projection approaches have been analyzed in the general case in [12].

The combination of learning and control techniques has been proposed in, e.g., [13]–[17]. The combination of RL and the linear quadratic regulator has been presented in [18] and [19]. To the best of our knowledge, [11], [20], and [21] are the first works proposing to use nonlinear MPC as a function approximator in RL. While strategies for providing some form of safety have been developed [8], to the best of the authors' knowledge, none of these approaches is able to strictly satisfy some set of constraints at all time. Rather, constraint violation is strongly penalized in [11] and some of the approaches in [8]. The only contribution providing robust constraint satisfaction guarantees is [22], where a linear feedback policy is learned.

In this article, we propose an RL formulation based on MPC that addresses the issue of safety, which we did not rigorously enforce in [11] and [20]. We summarize next two existing approaches and the scheme we propose, which combines them.

1) *Robust MPC, Business-as-usual:* A low-dimensional computationally tractable uncertainty set is first identified, and then, used to formulate a robust MPC problem [see (20)].
2) *RL, Business-as-usual:* Penalizing violations with a suitably high cost let the optimization procedure yield a policy that tends to not violate the constraints. Safety is typically not strictly guaranteed and only few results provide weak guarantees.

3) *Safe RL MPC:* In the approach we propose, a robust MPC problem is formulated similarly to 1). Similarly to 2), RL updates the parameterization of the robust MPC scheme, and of the safety constraint to reduce conservatism while preserving the safety.

Safe RL-MPC is based on the approach first advocated in [11] and [20]. The scheme can be seen from the following two alternative points of view: 1) MPC is used as a function approximator within RL in order to provide safety and stability guarantees and 2) RL is used in order to tune the MPC parameters, thus improving closed-loop performance in a data-driven fashion. Since safety is fundamental not only during exploitation, but also during exploration, we also address the issue of guaranteeing constraint satisfaction during this phase.

Another important contribution of this article is the development of an efficient way to deal with the enormous amount of data typically collected by autonomous systems. In particular, the introduction of a nominal (potentially inaccurate) linear model allows one to significantly reduce the amount of stored data. Further efficiency is obtained by exploiting convexity and using a low-dimensional approximation of the uncertainty set.

We first present RL at a conceptual level and propose adaptations in order to make RL applicable to the safety-enforcing setup. The main issue to be tackled is related to the safety constraints, which need to be enforced when updating the function approximator parameter. We formulate the update by resorting to a constrained optimization problem, similarly to what has been proposed in [20]. The proposed safe RL can be directly applied to Q-learning, but actor-critic techniques require some adaptation when the input space is continuous and restricted by safety requirements.

Contributions: This article proposes an approach to combine MPC and RL so as to guarantee safety. In order to limit the complexity of the algorithm, we rely on a linear system with bounded disturbances for predictions, and propose an *ad hoc* formulation of the robust constraint satisfaction that relies on the linear model to greatly reduce the amount of data to be stored. Finally, adaptations to the standard RL algorithms to guarantee that the parameter update does not jeopardize safety are introduced.

This article is structured as follows. We introduce the problem of safe RL in general terms in Section II, while the rest of this article specializes on the case of linear systems. We propose a tailored function approximator based on robust MPC in Section III and discuss the efficient use of data in Section IV. The necessary modifications to the standard RL algorithms are proposed in Section V and the whole framework is tested in simulations in Section VI. Finally, Section VII concludes this article with an outline for future research.

Notation: a is scalar, $\mathbf{a} \in \mathbb{R}^{n_\mathbf{a}}$ is a vector with components $\mathbf{a}_i$, A is a matrix with rows A_i, and $\mathbf{A}$ and $\mathcal{A}$ are sets. For any set, $|\cdot|$ defines its cardinality. For any function $\mathbf{f}(\mathbf{x})$, we define $\mathbf{f}(\mathbf{X}) := \{\, \mathbf{f}(\mathbf{x}) \,|\, \mathbf{x} \in \mathbf{X} \,\}$ and denote $\mathbf{f}(\mathbf{x}) \leq 0$, $\forall\, \mathbf{x} \in \mathbf{X}$ as $\mathbf{f}(\mathbf{X}) \leq 0$. The only exceptions are J, V, and Q, which are scalar functions, but denoted by capital letters in the literature.

II. Background and Safe RL Formulation

In this article, we consider real system dynamics described as a Markov process (MP) with continuous state $\mathbf{s}$ and action $\mathbf{a}$, with state transitions $\mathbf{s}, \mathbf{a} \to \mathbf{s}_+$ having the underlying conditional probability density

$$\mathbb{P}\left[\mathbf{s}_+ \,|\, \mathbf{s}, \mathbf{a}\right]. \tag{1}$$

We furthermore consider a deterministic policy delivering the control input as $\mathbf{a} = \boldsymbol{\pi}(\mathbf{s})$, resulting in state distribution $\tau^{\boldsymbol{\pi}}$. The RL problem then reads as

$$\boldsymbol{\pi}_\star := \arg\min_{\boldsymbol{\pi}} \; J(\boldsymbol{\pi}) := \mathbb{E}_{\tau^{\boldsymbol{\pi}}}\left[\sum_{k=0}^{\infty} \gamma^k \ell\left(\mathbf{s}_k, \boldsymbol{\pi}\left(\mathbf{s}_k\right)\right)\right] \tag{2}$$

where ℓ is called stage cost in optimal control and $-\ell$ instantaneous reward in RL. The scalar γ is a discount factor, typically smaller than 1 in RL, and 1 in MPC. Note that in (2), we provide the definition for nonepisodic settings, but the developments of this article readily apply to episodic settings. The value function $V_\star(\mathbf{s})$ is the optimal cost, obtained by applying the optimal policy $\boldsymbol{\pi}_\star$, i.e.,

$$V_\star(\mathbf{s}_0) = \mathbb{E}_{\tau^{\boldsymbol{\pi}_\star}}\left[\sum_{k=0}^{\infty} \gamma^k \ell\left(\mathbf{s}_k, \boldsymbol{\pi}_\star\left(\mathbf{s}_k\right)\right) \,\middle|\, \mathbf{s}_0\right] \tag{3}$$

and the Bellman equation defines the action-value function

$$Q_\star(\mathbf{s}, \mathbf{a}) := \ell(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}\left[\, V_\star(\mathbf{s}_+) \,|\, \mathbf{s}, \mathbf{a}\,\right] \tag{4}$$

where the expectation is taken over the state transition (1). The various forms of RL use parametric approximations V_θ, Q_θ, $\boldsymbol{\pi}_\theta$ of $V_\star$, $Q_\star$, and $\boldsymbol{\pi}_\star$ in order to find the parameter $\boldsymbol{\theta}^\star$ that (approximately) solves (2) either directly or by approximately solving (4) in a sampled-based fashion. For both approaches, we summarize this as

$$\boldsymbol{\theta}^\star := \min_{\boldsymbol{\theta}} \; \sum_{k=0}^{n} \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \tag{5}$$

where function ψ depends on the specific algorithm, e.g.,

$$\psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) = \left(\ell(\mathbf{s}_k, \mathbf{a}_k) + \gamma V_{\boldsymbol{\theta}_k}(\mathbf{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{s}_k, \mathbf{a}_k)\right)^2 \tag{6}$$

and $n = 1$ in recursive Q learning formulations and

$$\mathbb{E}_{\tau^{\boldsymbol{\pi}_\theta}}\left[\psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta})\right] = J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) \tag{7}$$

in policy gradient approaches [1]. To optimize performance, the system ought to be controlled by using the best available policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, this is referred to as *exploitation*. However, in order for problem (5) to be well-posed in general, it is necessary to also collect data by deviating from $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ and implementing a different policy $\boldsymbol{\pi}^{\mathrm{e}}$, this is referred to as *exploration*.

Among others, one of the main difficulties related with RL is safety enforcement, e.g., collision avoidance. In this article, we define safety through a set of constraints

$$\boldsymbol{\xi}(\mathbf{s}, \hat{\boldsymbol{\pi}}(\mathbf{s})) \leq 0 \qquad \forall\, \mathbf{s} \in \operatorname{supp}\left(\tau^{\hat{\boldsymbol{\pi}}}\right) \tag{8}$$

where we note $\operatorname{supp}(\tau^{\hat{\boldsymbol{\pi}}})$ the support of the distribution of the MP (1) subject to policy $\hat{\boldsymbol{\pi}}$. Ideally, condition (8) should

be satisfied at all times with unitary probability, both during exploitation, i.e., $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}_{\boldsymbol{\theta}}$, and exploration, i.e., $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}^{\mathrm{e}} \neq \boldsymbol{\pi}_{\boldsymbol{\theta}}$. Note that (8) can only hold if the process (1) has bounded support.

Enforcing (8) poses the following two major challenges:

1) either the support of (1) or the support of $\tau^{\hat{\boldsymbol{\pi}}}$ must be known or estimated;
2) given knowledge on either support, a policy satisfying (8) must be designed.

Problem 1) is fundamental, since one can never have the guarantee of being able to observe the full support of (1). Arguably, a reasonable approach can be to approximate the support based on the information extracted from the available samples, or on a prior, or on both. We assume that collected data are informative such that, in the limit for an infinite amount of data, the support is reconstructed exactly for policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. A theoretical justification of this is beyond the scope of this article. In order to construct an approximation of the MP support, we first define the dispersion set confining the state transitions as

$$\mathbf{S}_+ (\mathbf{s}, \mathbf{a}) = \{ \mathbf{s}_+ \mid \mathbb{P} [\mathbf{s}_+ \,|\, \mathbf{s}, \mathbf{a}] > 0 \}. \tag{9}$$

Since $\mathbf{S}_+$ is not known a priori, we introduce a parameterized approximation $\hat{\mathbf{S}}_+$ based on function $\mathbf{g}_{\boldsymbol{\theta}}$ given by

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) := \{ \mathbf{s}_+ \,|\, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{s}_+, \mathbf{s}, \mathbf{a}) \leq 0 \}. \tag{10}$$

In order to enforce safety, $\hat{\mathbf{S}}_+$ must be an outer approximation of set $\mathbf{S}_+$, i.e., $\boldsymbol{\theta}$ must be chosen such that

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) \supseteq \mathbf{S}_+(\mathbf{s}, \mathbf{a}) \qquad \forall\, \mathbf{s}, \mathbf{a}. \tag{11}$$

We label this condition *safe-design constraint* (SDC), since it restricts the values that $\boldsymbol{\theta}$ can take based on safety concerns. In order to discuss safety in mathematically simple terms, let us introduce the worst-case mass of (1) outside of $\hat{\mathbf{S}}_+$, defined as

$$\chi (\boldsymbol{\theta}) = \sup_{\mathbf{s}, \mathbf{a}} \mathbb{E} \left[\mathbf{s}_+ \notin \hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) \,\middle|\, \mathbf{s}, \mathbf{a} \right] \in [0, 1] \tag{12}$$

where the expected value is taken over (1). For a given $\boldsymbol{\theta}$, (11) is ensured if $\chi(\boldsymbol{\theta}) = 0$. However, $\chi(\boldsymbol{\theta})$ is known only if we assume the real system dynamics (1) are known. Otherwise, in the Bayesian context, $\chi(\boldsymbol{\theta})$ ought to be treated as a random variable, reflecting our imperfect knowledge of it, conditioned on the current data $\mathcal{D}$ (i.e., knowledge) we have of the system. We, therefore, consider

$$\eta (\mathcal{D}) = \mathbb{P} [\, \chi (\boldsymbol{\theta}) = 0 \,|\, \mathcal{D} \,] \tag{13}$$

which provides a formal definition of the probability that $\hat{\mathbf{S}}_+$ is capturing the real system dispersion provided the data $\mathcal{D}$. While computing η is difficult in the general case, its definition allows us to discuss in rigorous terms the fundamental limitation provided next.

Fundamental Limitation 1: For any $\mathcal{D}$, if $\hat{\mathbf{S}}_+ \subset \mathbb{R}^{n_{\mathrm{s}}}$, it is impossible to guarantee that

$$\eta (\mathcal{D}) = 1 \tag{14}$$

without introducing any additional assumption on (1).

In other words, given a finite set of samples, and possibly, prior but not absolute knowledge of the system, one does not have enough information to construct a set containing all future samples, unless this set is $\mathbb{R}^{n_{\mathrm{s}}}$. This is a fundamental limitation of robust constraint satisfaction, and therefore, it is independent of the proposed approach. In the following, we will, therefore, discuss η-safety, underlining that (14) cannot be achieved in practice.

Arguably, by introducing additional assumptions restricting the function space to which (1) can belong, one can envision overcoming this limitation. The investigation of this topic is beyond the scope of this article and will be the subject of future research.

In order to formulate some form of SDC with limited information, we introduce the sample-based form of (11) as

$$\mathbf{s}_{k+1} \in \hat{\mathbf{S}}_+(\mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \qquad \forall\, k. \tag{15}$$

For problem 2), the main challenge is to find values of $\boldsymbol{\theta}$ such that the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ strictly satisfies the safety constraints.

We define safety based on the dispersion set propagation under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, which reads as

$$\mathbf{S}_{k+1}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} := \hat{\mathbf{S}}_+ \left(\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}, \boldsymbol{\pi}_{\boldsymbol{\theta}} \left(\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \right), \boldsymbol{\theta} \right), \qquad \mathbf{S}_0^{\boldsymbol{\pi}_{\boldsymbol{\theta}}} = \mathbf{s}_0. \tag{16}$$

Definition 1 (η-safe Policy): For a given dataset $\mathcal{D}$ and a given set of initial conditions $\mathcal{S}_0$, a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is labeled as η-safe for initial states $\mathbf{s}_0 \in \mathcal{S}_0$ if it satisfies

$$\boldsymbol{\xi}(\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta}}})) \leq 0 \qquad \forall\, k \geq 0. \tag{17}$$

In general, there can exist initial states for which a safe policy cannot exist [23], and Definition 1 characterizes policies that preserve safety for a given initial state.

Providing safety guarantees is arguably an open problem when using DNNs as function approximators. However, this problem has been studied in control theory and one successful design technique is robust MPC [24]–[26]. Therefore, instead of building the function approximations based on the commonly used DNN approaches, we will use robust MPC, within an extended version of the RL-MPC scheme proposed in [11] and [20]. Note that it has been proven in [11] that the optimal policy $\boldsymbol{\pi}$, value, and action-value functions $V_\star$ and $Q_\star$ can be recovered exactly by function approximations based on MPC, provided that their parameterization is rich enough, even in case the model used in MPC is different from (1).

In order to guarantee safety, robust MPC would ideally rely on the propagation of the dispersion set (16). Unfortunately, this poses severe computational challenges and an auxiliary (time varying) policy $\boldsymbol{\pi}_{\boldsymbol{\theta},k}^{\mathrm{MPC}}$ is preferred in order to recover a computationally tractable formulation [25]. Policy $\boldsymbol{\pi}_{\boldsymbol{\theta},k}^{\mathrm{MPC}}$ is typically selected as an open-loop input profile corrected by an affine feedback (see Section III-A). We remark that MPC delivers $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}_{\boldsymbol{\theta},0}^{\mathrm{MPC}}$. By construction, robust MPC delivers a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ that satisfies constraints $\boldsymbol{\xi}$ at all future times, provided that the SDC (15) holds for all k. Even though the dispersion set propagation $\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta},k}^{\mathrm{MPC}}}$ is computed based on $\boldsymbol{\pi}_{\boldsymbol{\theta},k}^{\mathrm{MPC}}$, the constraints are guaranteed to hold also for $\mathbf{S}_k^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$.

Since the shape of $\hat{\mathbf{S}}_+$ impacts the closed-loop performance, one can let safe RL adapt $\hat{\mathbf{S}}_+$. However, this requires one to enforce (15) explicitly in RL. The safe RL problem is then

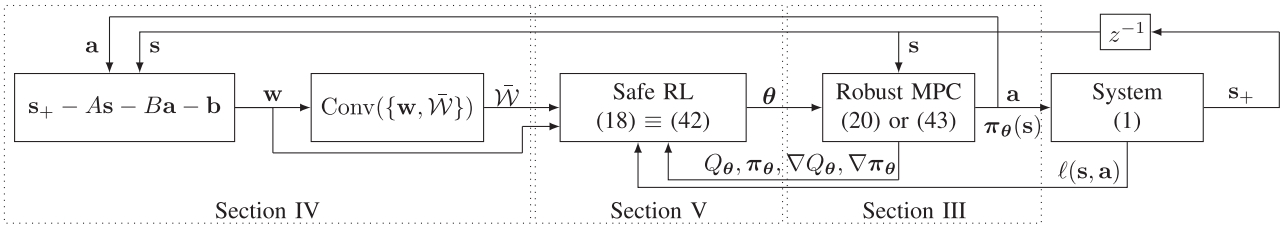

Fig. 1. Schematics of the proposed setup: Data are used to construct the SDC based on $\bar{\mathcal{W}}$ and to evaluate the cost in (5). This cost depends on $Q_{\boldsymbol{\theta}}, V_{\boldsymbol{\theta}}$ obtained from MPC, and ℓ. MPC controls the system. The signal toggling between exploitation and exploration is omitted to avoid confusion, and is a signal sent from RL to switch between MPC (20) and (43).

formulated as

$$\boldsymbol{\theta}^\star := \min_{\boldsymbol{\theta}} \sum_{k=0}^{n} \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \tag{18a}$$

$$\text{s.t. } \mathbf{s}_{k+1} \in \hat{\mathbf{S}}_+(\mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \quad \forall\, k. \tag{18b}$$

By enforcing (18b), RL is explicitly made aware that some parameter updates are unsafe, and therefore, not feasible. Provided that the SDC holds, MPC delivers a safe policy by construction. Though in principle the SDC has to be enforced for each sample, in Section IV, we propose an approach to largely reduce the amount of constraints.

Remark 1: The RL problem (18) is typically solved using sensitivity-based methods, hence, we need to differentiate the function approximator with respect to the parameter $\boldsymbol{\theta}$. In our case, we need to differentiate the robust MPC problem. This will be detailed in Section III-B.

The proposed safe RL framework performs the following steps:

1) at every time instant, MPC is solved and differentiated; the MPC input is applied to the system; state transitions are observed and data are collected to form the sample based SDC (15);
2) the RL problem (18) is solved (possibly at a lower sampling rate than MPC) and parameter $\boldsymbol{\theta}$ is updated whenever possible.

A scheme is displayed in Fig. 1 , with reference to the section where each component is discussed.

In this section, we have established the safe RL framework; in the next sections, we will discuss the following:

1) how to implement robust MPC;
2) how to differentiate it in order to be able to solve problem (18);
3) how to manage constraint (18b) in a data-efficient fashion.

In this article, we address 1)–3) by relying on a linear model of (1) to enforce safety. On the one hand, this choice makes the robust constraint satisfaction tractable and not excessively demanding in terms of computations. On the other hand, any nonlinearity present in the system will be accounted for as a perturbation, therefore, introducing some conservatism. We remark that nonlinear robust MPC formulations have been developed and can be deployed within the proposed algorithmic framework. The main drawbacks of these formulations are as follows:

1) some form of conservatism cannot be avoided;
2) the computational burden typically becomes prohibitive.

III. Robust MPC Based on Invariant Sets

Since guaranteeing robust constraint satisfaction in the general nonlinear case is extremely difficult [25], in the remainder of this article, we focus on the case of an affine model. We describe safety constraints (8) as the inner approximation

$$C\mathbf{s} + D\mathbf{a} + \bar{\mathbf{c}} \leq \mathbf{0}. \tag{19}$$

We formulate a Q function approximator based on the classic robust linear MPC [24], [25] as

$$Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) :=$$

$$\min_{\mathbf{z}} \sum_{k=0}^{N-1} \gamma^k \left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top H \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \mathbf{h}^\top \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \right) + \gamma^N \left(\mathbf{x}_N^\top P \mathbf{x}_N + \mathbf{p}^\top \mathbf{x}_N \right) \tag{20a}$$

$$\text{s.t. } \mathbf{x}_0 = \mathbf{s}, \quad \mathbf{u}_0 = \mathbf{a}, \tag{20b}$$

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{b}, \quad k \in \mathbb{I}_0^{N-1} \tag{20c}$$

$$C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \leq \mathbf{0}, \quad k \in \mathbb{I}_0^{N-1} \tag{20d}$$

$$G\mathbf{x}_N + \mathbf{g} \leq \mathbf{0} \tag{20e}$$

where $\mathbf{z} := (\mathbf{x}_0, \mathbf{u}_0, \ldots, \mathbf{u}_{N-1}, \mathbf{x}_N)$. Note that we use $\mathbf{x}$, $\mathbf{u}$ to distinguish the MPC predictions from the actual state and action realizations $\mathbf{s}, \mathbf{a}$, which define the initial constraint (20b). The dynamic constraints (20c) assume a nominal model without any perturbation. The tube-based approach then treats the system stochasticity, model uncertainties, and safety constraint (19) by performing a suitable tightening of the path constraints (20d), i.e., $\mathbf{c}_k \geq \bar{\mathbf{c}}$ is used. The terminal constraints (20e) are introduced to guarantee that the path constraints will never be violated at all future times $k > N$.

The value function and optimal policy are obtained as [11]

$$V_{\boldsymbol{\theta}}(\mathbf{s}) := \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \quad \pi_{\boldsymbol{\theta}}(\mathbf{s}) := \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}). \tag{21}$$

In practice, $V_{\boldsymbol{\theta}}$ and $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ are computed jointly by solving (20) without enforcing the constraint $\mathbf{u}_0 = \mathbf{a}$. Parameter $\boldsymbol{\theta}$ to be adapted by RL may include any of the vector and matrices defining the MPC scheme (20), i.e., generally

$$\boldsymbol{\theta} = \{H, \mathbf{h}, P, \mathbf{p}, A, B, \mathbf{b}, \bar{\mathbf{c}}, K, \boldsymbol{\theta}_{\mathbf{W}}\}. \tag{22}$$

Parameters H and P are typically assumed positive definite to guarantee the solvability of (20). Parameters K and $\boldsymbol{\theta}_{\mathbf{W}}$ do not appear explicitly in (20), and are used to compute the constraint tightening and the terminal set, which will be introduced next: we will first present the computation of the constraint tightening, and then, discuss the computation of the sensitivities of $V_{\boldsymbol{\theta}}$, $Q_{\boldsymbol{\theta}}$, and $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$.

A. Recursive Robust Constraint Satisfaction

In order to guarantee constraint satisfaction for the real system (1) using predictions given by the nominal model (20c), robust MPC explains the difference between predictions and actual state transitions by means of additive noise $\mathbf{w} \in \mathbf{W}_{\boldsymbol{\theta}}$, with $\mathbf{W}_{\boldsymbol{\theta}}$ satisfying

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) = A\mathbf{s} + B\mathbf{a} + \mathbf{b} + \mathbf{W}_{\boldsymbol{\theta}} \supseteq \mathbf{S}_+(\mathbf{s}, \mathbf{a}). \tag{23}$$

We remark that, by using the affine model (20c), conservatism is introduced as any nonlinearity present in (1) will be accounted for by $\mathbf{w}$. Set $\mathbf{W}_{\boldsymbol{\theta}}$ is parameterized by parameter $\boldsymbol{\theta}_{\mathbf{W}}$. Common choices in robust MPC are to parameterize $\mathbf{W}_{\boldsymbol{\theta}}$ using ellipsoids or polytopes; in this article, we consider the latter.

In order to perform constraint tightening, i.e., the computation of $\mathbf{c}_k$, we rely on the approach first proposed by [24]. We introduce the prediction error of the nominal model (20c) as

$$\mathbf{E}_{k+1} = (A - BK)\mathbf{E}_k + \mathbf{W}_{\boldsymbol{\theta}}, \qquad \mathbf{E}_0 = \{\mathbf{0}\}$$

where set $\mathbf{E}_k$ predicts an outer approximation of the dispersion set around the predicted trajectory, i.e., $\mathbf{S}_k^{\boldsymbol{\pi}^{\mathrm{MPC}}_{\boldsymbol{\theta},k}} \subseteq \mathbf{x}_k + \mathbf{E}_k$, $k = 0, \ldots, N$, where $\boldsymbol{\pi}^{\mathrm{MPC}}_{\boldsymbol{\theta},k} := \mathbf{a}_k - K\mathbf{e}_k$, $\mathbf{e}_k \in \mathbf{E}_k$ and

$$\mathbf{S}_{k+1}^{\boldsymbol{\pi}^{\mathrm{MPC}}_{\boldsymbol{\theta}}} = \hat{\mathbf{S}}_+(\mathbf{S}_k^{\boldsymbol{\pi}^{\mathrm{MPC}}_{\boldsymbol{\theta},k}}, \mathbf{a}_k - K\mathbf{E}_k, \boldsymbol{\theta}), \quad \mathbf{S}_0^{\boldsymbol{\pi}^{\mathrm{MPC}}_{\boldsymbol{\theta},k}} = \{\mathbf{s}_k\}.$$

The feedback matrix K is introduced in order to model the fact that any closed-loop strategy will compensate for perturbations on the nominal model. For ease of notation, we define

$$C_K := C - DK, \quad A_K := A - BK.$$

Robust constraint satisfaction is then obtained provided that

$$C\mathbf{x}_k + D\mathbf{u}_k + C_K\mathbf{e}_k + \bar{\mathbf{c}} \le \mathbf{0} \qquad \forall\, \mathbf{e}_k \in \mathbf{E}_k \tag{24}$$

such that $\mathbf{c}_k$ is obtained by adding the worst-case realization of $C_K\mathbf{E}_k$ to $\bar{\mathbf{c}}$. For each component i of the path constraint at time k, we define

$$\begin{aligned}\mathbf{d}_{i,k} &:= \max_{\mathbf{e}}\ (C_K)_i\mathbf{e} && \text{s.t. } \mathbf{e} \in \mathbf{E}_k \\ &= \max_{\mathbf{w}}\ (C_K)_i \sum_{j=0}^{k-1} (A_K)^j \mathbf{w}_j && \text{s.t. } \mathbf{w}_j \in \mathbf{W}_{\boldsymbol{\theta}}. \end{aligned} \tag{25}$$

We lump all components $\mathbf{d}_{i,k}$ in vector $\mathbf{d}_k$. Then, constraint satisfaction is obtained for all $\mathbf{w}_k \in \mathbf{W}_{\boldsymbol{\theta}}$ if

$$\mathbf{c}_k = \bar{\mathbf{c}} + \mathbf{d}_k.$$

If $\mathbf{W}_{\boldsymbol{\theta}}$ is a polytope, then (25) can be formulated as a linear program (LP); this implies that constraint tightening is relatively cheap to compute; and as detailed in Section IV, the SDC enforcement becomes easier to derive.

In order to guarantee that problem (20) remains feasible at all times for all $\mathbf{w}_k \in \mathbf{W}_{\boldsymbol{\theta}}$, one needs to impose *ad hoc* terminal conditions. More specifically, the terminal set $\mathcal{X}_{\mathrm{f}} := \{\, \mathbf{x} \,|\, G\mathbf{x} + \mathbf{g} \le 0 \,\}$ should be robustly invariant and output admissible, i.e., there exists a terminal control law $\boldsymbol{\kappa}_{\mathrm{f}}$ such that $\mathbf{s}_{k+1} \in \mathcal{X}_{\mathrm{f}}$ and $C\mathbf{s}_k + D\boldsymbol{\kappa}_{\mathbf{f}}(\mathbf{s}_k) + \bar{\mathbf{c}} \le \mathbf{0}$ for every $\mathbf{s}_k \in \mathcal{X}_{\mathrm{f}}$. We consider a linear control law $\boldsymbol{\kappa}_{\mathrm{f}}(\mathbf{s}) = -K\mathbf{s}$, coinciding with the linear feedback used to stabilize the prediction error $\mathbf{e}$.

In order to compute set $\mathcal{X}_{\mathrm{f}}$, we define

$$\begin{aligned}\mathcal{X}_0 &:= \{\, \mathbf{x} \,|\, C_K\mathbf{x} + \mathbf{c}_0 \le 0 \,\} \\ \mathcal{X}_k &:= \{\, \mathbf{x} \in A_K\mathcal{X}_{k-1} \oplus \mathbf{W}_{\boldsymbol{\theta}} \,|\, C_K\mathbf{x} + \mathbf{c}_k \le 0 \,\}.\end{aligned}$$

Note that, by (24) and (25), $\mathbf{x}_0 \in \mathcal{X}_k$ implies

$$C_K(\mathbf{x}_j + \mathbf{e}_j) + \bar{\mathbf{c}} \le 0 \qquad \forall\, \mathbf{e}_j \in E_j, \qquad \forall\, i \in \mathbb{I}_0^k.$$

Set $\mathcal{X}_\infty$ is maximal robust positive invariant (MRPI) [27]

$$\mathbf{s}_k \in \mathcal{X}_\infty \ \Rightarrow\ A_K^{j-k}\mathbf{s}_k \in \mathcal{X}_\infty,\ C_K\mathbf{s}_j + \bar{\mathbf{c}} \le 0\ \forall\, j > k.$$

Additionally, whenever the system is stable and the origin is in the interior of the constraint set, the MRPI set is finitely determined [27, Th. 6.3], i.e., $\exists\, k' < \infty$ s.t. $\mathcal{X}_{k'} \equiv \mathcal{X}_{k'+i}$, for all $i \ge 1$. The stability requirement further motivates the introduction of feedback through the matrix K.

As proven in [27], for $\mathbf{W}_{\boldsymbol{\theta}}$ polyhedral, G, $\bar{\mathbf{g}}$ are given by

$$G := \begin{bmatrix} C_K \\ C_K A_K \\ \vdots \\ C_K A_K^{k'} \end{bmatrix}, \qquad \bar{\mathbf{g}} := \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{k'} \end{bmatrix} \tag{26}$$

where we stress that $\mathbf{c}_k = \bar{\mathbf{c}} + \mathbf{d}_k$, such that, by using $\boldsymbol{\kappa}_{\mathrm{f}} = -K\mathbf{e}$, one can spare a large amount of computations, $\bar{\mathbf{g}}$ being already computed. The condition $G\mathbf{x}_N + \bar{\mathbf{g}} \le 0$ would then guarantee robust constraint satisfaction for all future times if $\mathbf{s}_N = \mathbf{x}_N$. However, $\mathbf{s}_N = \mathbf{x}_N + \mathbf{e}_N$, such that also the terminal constraint (20e) must be tightened. Analogously to the case of path constraints, we define $\mathbf{g} = \bar{\mathbf{g}} + \mathbf{h}$ with

$$\mathbf{h}_{i,k} := \max_{\mathbf{w}}\ G_i \sum_{j=0}^{k'-1} A_K^j \mathbf{w}_j \qquad \text{s.t. } \mathbf{w}_j \in \mathbf{W}_{\boldsymbol{\theta}}. \tag{27}$$

Remark 2: Typically, constraints that can never become active are removed from (26), so as to reduce the dimension.

Safety is guaranteed by the following result on tube MPC.

Proposition 1 (Recursive Feasibility): Assume that $\mathcal{X}_{\mathrm{f}} := \{\, \mathbf{x} \,|\, G\mathbf{x} + \mathbf{g} \le 0 \,\}$ is robust positive invariant (RPI) and problem (20) is feasible at time $k = 0$. Then, problem (20) is feasible for all $\mathbf{w}_k \in \mathbf{W}_{\boldsymbol{\theta}}$ and all times $k \ge 0$. If moreover (23) holds, the real system (1) satisfies the safety constraint (19) at all times $k \ge 0$.

Proof: The proof can be found in, e.g., [24]. ■

This proof is valid in case $\boldsymbol{\theta}$ is kept fixed. A discussion on how to enforce recursive feasibility upon updates of $\boldsymbol{\theta}$ is proposed in Section V, Remark 9.

While the framework of robust linear MPC based on MRPI sets is well established, the computation of the parametric sensitivities of an MPC problem required to deploy most RL

methods is not common. In particular, in the case of a tube-based formulation, also the constraint tightening procedure needs to be differentiated. This is also not common and deserves to be discussed in detail. We, therefore, devote the next subsection to the computation of the derivative of the value and action-value function with respect to parameter $\boldsymbol{\theta}$.

B. Differentiability

In order to be able to deploy RL algorithms to adapt parameter $\boldsymbol{\theta}$, we need to be able to differentiate the MPC scheme, and therefore, the constraint definition with respect to $\boldsymbol{\theta}$. In principle, $\boldsymbol{\theta}$ could include $A, B,$ and K, but also all other parameters of the MPC formulation (22). In order to compute $\nabla_{\boldsymbol{\theta}} \mathbf{c}_k$, $\nabla_{\boldsymbol{\theta}} C_N$, and $\nabla_{\boldsymbol{\theta}} \mathbf{c}_N$, one can use results from parametric optimization to obtain the following lemmas [28].

Consider a parametric NLP with cost $\phi_{\boldsymbol{\theta}}$, primal-dual variable $\mathbf{y}$ and parameter $\boldsymbol{\theta}$. We refer to [29] for the definition of Lagrangian $l^0_{\boldsymbol{\theta}}(\mathbf{y})$, KKT conditions, linear independence constraint qualification (LICQ), second-order sufficient conditions (SOSC), and strict complementarity (SC). For a fixed active set, the KKT conditions reduce to the equality $\mathbf{r}^0_{\boldsymbol{\theta}}(\mathbf{y}) = 0$.

Lemma 1: Consider a parametric optimization problem with optimal primal-dual solution $\mathbf{y}^\star$. Assume that LICQ, SOSC, and SC hold at $\mathbf{y}^\star$. Then, the following holds:

$$\nabla_{\boldsymbol{\theta}} \phi_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} l^0_{\boldsymbol{\theta}}(\mathbf{y}), \qquad \frac{\partial \mathbf{r}^0_{\boldsymbol{\theta}}}{\partial \mathbf{y}} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \mathbf{y}^\star = \frac{\partial \mathbf{r}^0_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}.$$

Proof: The result can be found in, e.g., [28]. ■

Corollary 1: Assume that LICQ, SOSC, and SC hold at the optimal solution of (20). Then, the value function $V_{\boldsymbol{\theta}}$, action-value function $Q_{\boldsymbol{\theta}}$, and optimal solution $\mathbf{y}^\star$ (therefore, also policy $\boldsymbol{\pi}$) are differentiable with respect to parameter $\boldsymbol{\theta}$, with

$$\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(\mathbf{s}) = \nabla_{\boldsymbol{\theta}} \bar{l}_{\boldsymbol{\theta}}(\mathbf{y}), \qquad \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \nabla_{\boldsymbol{\theta}} l_{\boldsymbol{\theta}}(\mathbf{y}) \tag{28a}$$

$$\frac{\partial \mathbf{r}_{\boldsymbol{\theta}}}{\partial \mathbf{y}} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \mathbf{y}^\star = \frac{\partial \mathbf{r}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \tag{28b}$$

where $l_{\boldsymbol{\theta}}$ is the Lagrangian of the problem (20), $\bar{l}_{\boldsymbol{\theta}}$ is the Lagrangian when constraint $\mathbf{u}_0 = \mathbf{a}$ is eliminated, and $\mathbf{r}_{\boldsymbol{\theta}}$ denotes the Karush–Kuhn–Tucker (KKT) conditions for the optimal active set.

Remark 3: When solving an LP, QP, or NLP using a second-order method, e.g., active set or interior point, the most expensive operation is the factorization of the KKT matrix $\frac{\partial \mathbf{r}_{\boldsymbol{\theta}}}{\partial \mathbf{y}}$. Once the matrix is factorized, the solution of the linear system is computationally inexpensive. Therefore, the sensitivities of the solution are in general much cheaper to evaluate than solving the problem itself. The sensitivity of the optimal value function is even simpler to compute, since it consists in the differentiation of the Lagrangian [see (28a)].

In (28), $\mathbf{r}_{\boldsymbol{\theta}}$, $l_{\boldsymbol{\theta}}$, and $\bar{l}_{\boldsymbol{\theta}}$ depend on $\mathbf{c}_k$ and $\mathbf{g}$ which, in turn, depend on $\boldsymbol{\theta}$ as they are optimal values of parametric optimization problems (25) and (27). Consequently, one needs to evaluate $\nabla_{\boldsymbol{\theta}} \mathbf{c}_k$ and $\nabla_{\boldsymbol{\theta}} \mathbf{g}$. In the following, we further detail the application of Lemma 1 to this case.

We consider only problem (25), since the derivation for (27) is analogous. First, we state separability and, therefore, parallelizability of the computation of $\mathbf{d}_k$ in the following Lemma.

Lemma 2: Each component of $\mathbf{d}_k$ can be computed as

$$\mathbf{d}_{i,k} = \sum_{j=0}^{k-1} \mathbf{d}_{i,k,j}, \text{ where}$$

$$\mathbf{d}_{i,k,j} := \max_{\mathbf{w}_j} (C_K)_i A_K^j \mathbf{w}_j \qquad \text{s.t.} \qquad \mathbf{w}_j \in \mathcal{W}_{\boldsymbol{\theta}}. \tag{29}$$

Proof: Each term in the sum $\sum_{j=0}^{k-1} A_K^j \mathbf{w}_j$ depends only on variable $\mathbf{w}_j$, and the problem is fully separable. ■

Then, Lemma 1 can be applied to obtain

$$\frac{\mathrm{d}\mathbf{d}_k}{\mathrm{d}\boldsymbol{\theta}} = \left(\frac{\mathrm{d}\mathbf{d}_{1,k}}{\mathrm{d}\boldsymbol{\theta}}, \ldots, \frac{\mathrm{d}\mathbf{d}_{n_{\mathbf{c}_k},k}}{\mathrm{d}\boldsymbol{\theta}} \right)$$

$$\frac{\mathrm{d}\mathbf{d}_{i,k}}{\mathrm{d}\boldsymbol{\theta}} = \sum_{j=0}^{k-1} \frac{\mathrm{d}\mathbf{d}_{i,k,j}}{\mathrm{d}\boldsymbol{\theta}}, \qquad \frac{\mathrm{d}\mathbf{d}_{i,k,j}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}l^{\mathbf{d}}_{\boldsymbol{\theta},i,j}}{\mathrm{d}\boldsymbol{\theta}}$$

where $l^{\mathbf{d}}_{\boldsymbol{\theta},i,j}$ is the Lagrangian of the problem (29). Provided that a second-order method is used for solving problem (29), then the matrix factorization is available and can be reused to compute the sensitivities at a negligible cost. For any function $\mathbf{f}(\mathbf{c}_k(\boldsymbol{\theta}))$, the chain rule yields

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{c}_k} \frac{\mathrm{d}\mathbf{c}_k}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{c}_k} \frac{\mathrm{d}\mathbf{d}_k}{\mathrm{d}\boldsymbol{\theta}}.$$

Remark 4: As underlined previously, problem (29) can be solved in parallel not only for each prediction time k, but also for each component i. Moreover, if an active-set solver is used, the active set can be initialized and the matrix factorization reused, such that often there will be no need for recomputing the factorization and computations can be done in an extremely efficient manner. Finally, problems (29) are very low dimensional and are, therefore, solved extremely quickly.

C. Guaranteeing MPC Feasibility and LICQ

We detail next two issues that can be easily encountered when deploying RL based on MPC (20). Since these are common, a simple solution that has become a standard in MPC is readily available.

1) MPC Feasibility: Since the set of possible perturbations is not known *a priori* but rather approximated as $\mathbf{W}_{\boldsymbol{\theta}}$ based on the collected samples, it cannot be excluded that some future sample $\mathbf{w}_k$ will not be inside the set, i.e., $\mathbf{w}_k \notin \mathbf{W}_{\boldsymbol{\theta}}$. The set approximation must then be modified to include the new sample [see Section IV-B and (40)], but recursive feasibility is potentially lost, no action is computed and the controller stops working.

2) Sensitivity Computation: The sensitivity computation is valid only if LICQ holds. However, problem (20) is not guaranteed to satisfy LICQ.

We propose to address both issues by using a common approach in MPC, i.e., a constraint relaxation for constraints (20d) and (20e) with an exact penalty [30]: variables $\boldsymbol{\sigma}_k$ are introduced, the constraints are modified as

$$C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \leq \boldsymbol{\sigma}_k \tag{30a}$$

$$G\mathbf{x}_N + \mathbf{g} \leq \boldsymbol{\sigma}_N, \qquad \boldsymbol{\sigma}_k \geq 0, \qquad k \in \mathbb{I}_1^N \tag{30b}$$

and the term $\sum_{k=1}^{N} \boldsymbol{\rho}^\top \boldsymbol{\sigma}_k$ is added to the cost.

Remark 5: Constraints only involving the controls do not need to be relaxed, while any constraint involving the states should not be imposed at $k = 0$, since then LICQ cannot be guaranteed even if the relaxation proposed previously is deployed.

We formalize the result in the next proposition.

Proposition 2: Assume that constraints (20d) and (20e) are relaxed as per (30) and the term $\sum_{k=1}^{N} \boldsymbol{\rho}^\top \boldsymbol{\sigma}_k$ is added to the cost. Then, if $\boldsymbol{\rho} < \infty$ is large enough, the solution is unchanged whenever feasible and recursive feasibility and LICQ are guaranteed.

Proof: The first result, i.e., solution equivalence whenever feasible and recursive feasibility is well known (see [30] and [31, Th. 14.3.1]). Regarding LICQ, we first note that in a formulation without constraints (20d) and (20e) LICQ holds by construction: Constraints (20b) and (20c) can be eliminated by condensing [32], yielding a problem with $Nn_{\mathbf{u}}$ unconstrained variables. The introduction of any linearly independent set of pure control constraints does then not jeopardize LICQ by construction. Assume now to introduce a linearly dependent constraint of the form (20d) or (20e) with Jacobian $\boldsymbol{\nu}$. By introducing slack variable σ, the new Jacobian becomes $[\boldsymbol{\nu}\ 1]$, which is by construction linearly independent with $[\boldsymbol{\nu}\ 0]$, and consequently, with the other constraints in the problem. ■

As discussed in Section II, safety holds with probability $\eta(\mathcal{D})$. This is an intrinsic issue of any safety-enforcing control scheme, and the proposed relaxation only solves the issue of avoiding infeasibility for the MPC scheme. This is further discussed in Section V-B.

IV. Safe Design Constraint and Data Management

As explained in the previous section, safety is obtained if all possible state transitions are correctly captured in the MPC formulation, i.e., if $\mathbf{W}_{\boldsymbol{\theta}}$, and therefore, $\mathbf{g}_{\boldsymbol{\theta}}$ is correctly identified. In other words, based on the collected data, the SDC must be enforced in order to guarantee that the uncertainty described by $\mathbf{W}_{\boldsymbol{\theta}}$ is representative of the real system (1). In this section, we propose a sample-based formulation of the SDC (11) to be used within RL formulations (5).

Many control systems are typically operated at high sampling rates, and consequently, data are collected at high rates. In order to be able to deal in real time with the large amounts of data that cumulate, it is necessary to retain only strictly relevant data, and compress the available information using appropriately defined data structures. In the following, we first discuss the data structures involved in RL-MPC, and then, discuss how to make an efficient use of data in the context of the proposed MPC formulation.

A. Set Membership and SDC

Consider the (possibly very large) set of state transitions observed on the real system

$$\mathcal{D} = \{(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2), \ldots, (\mathbf{s}_n, \mathbf{a}_n, \mathbf{s}_{n+1})\}. \tag{31}$$

The problem of enforcing the SDC (15) is related to the one of estimating the dispersion set $\mathbf{S}_+$ from data, which has been studied in the context of *set-membership system identification* (SMSI) [33]. Essentially, the dispersion set must satisfy

$$\mathbf{s}_+ \in \hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) \qquad \forall\ (\mathbf{s}, \mathbf{a}, \mathbf{s}_+) \in \mathcal{D} \tag{32}$$

i.e., $\boldsymbol{\theta}$ must satisfy the SDC (15), and therefore, belongs to set

$$\mathcal{S}_{\mathcal{D}} := \{\boldsymbol{\theta} \,|\, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{u}_k) \leq 0 \,\forall\, (\mathbf{s}_+, \mathbf{s}, \mathbf{u}) \in \mathcal{D}\}. \tag{33}$$

We should underline here the difference between $\hat{\mathbf{S}}_+$ defined in (10) and $\mathcal{S}_{\mathcal{D}}$. The former is an outer approximation of the set of all realizations $\mathbf{s}_+$, given state and action $\mathbf{s}, \mathbf{a}$, parameterized by $\boldsymbol{\theta}$. The latter instead describes the set of parameters $\boldsymbol{\theta}$ such that all state transitions from $\mathcal{D}$ are contained in $\hat{\mathbf{S}}_+$.

In SMSI, the parameter $\boldsymbol{\theta}$ is typically selected so as to obtain the smallest possible set $\mathbf{S}_+$. In the absence of specific information on the control task to be executed, this is arguably a very reasonable approach. However, given a specific control task to be executed, better performance might be obtained by selecting parameter $\boldsymbol{\theta}$ to approximate some part of $\mathbf{S}_+$ accurately even at the cost of increasing the volume of $\hat{\mathbf{S}}_+$. We, therefore, provide the following definition.

Definition 2. (Set Membership Optimality): Equivalently to (2), we define the closed-loop cost $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ associated with the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ learned by RL when relying on the set $\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta})$. Then, parameter $\boldsymbol{\theta}$ is (locally) optimal for the control task iff $\nexists\ \bar{\boldsymbol{\theta}} \in \mathcal{B}_\epsilon(\boldsymbol{\theta})$ s.t. $J(\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}) < J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$, where $\mathcal{B}_\epsilon(\cdot)$ denotes a ball of radius ϵ centered at $\boldsymbol{\theta}$.

In other words, the optimal parameter minimizes the cost subject to the SDC (15). Based on this definition, the optimal set approximation is obtained by an SMSI that selects $\boldsymbol{\theta}$ to maximize the closed-loop performance of the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. In this article, we aim at doing so by means of RL.

Every time the RL problem (18) is solved, one must ensure that $\boldsymbol{\theta} \in \mathcal{S}_{\mathcal{D}}$, i.e., $|\mathcal{D}|$ constraints need to be included in the problem formulation. With large amounts of data, this could make the problem computationally intractable. In the following, we will analyze how to tackle this issue.

While the previous developments did not require any assumption on function $\mathbf{g}_{\boldsymbol{\theta}}$, in order to efficiently manage data, we will assume hereafter that $\mathbf{g}_{\boldsymbol{\theta}}$ is affine in $\mathbf{s}, \mathbf{a}$. Note that this choice is consistent with the choice of an affine model. The SDC then becomes

$$\mathcal{S}_{\mathcal{D}} := \{\boldsymbol{\theta} \,|\, M\,(\mathbf{s}_+ - A\mathbf{s} - B\mathbf{a} - \mathbf{b}) \leq \mathbf{m}\ \forall\, (\mathbf{s}_+, \mathbf{s}, \mathbf{u}) \in \mathcal{D}\}$$

for some $M, \mathbf{m}$ possibly part of $\boldsymbol{\theta}$. Since both $\mathcal{S}_{\mathcal{D}}$ and $\mathbf{W}_{\boldsymbol{\theta}}$ are defined based on $\mathbf{g}_{\boldsymbol{\theta}}$, parameters M and $\mathbf{m}$ for the two sets coincide. Therefore, the SDC directly defines the uncertainty set $\mathbf{W}_{\boldsymbol{\theta}}$ used by MPC to compute safe policies. Note that A, B, $\mathbf{b}$, M, and $\mathbf{m}$ can all be included in $\boldsymbol{\theta}$, and therefore, adapted

by RL. Which parameter to adapt or keep constant is a design choice (see also Remark 7).

B. Model-Based Data Compression

In this section, we discuss how to compress the available data without loss of information to significantly reduce the complexity of safe RL. It will become clear that the nominal model (20c) plays a very important role in this context, as it makes it possible to organize data such that it can be efficiently exploited. Unfortunately, this efficiency is lost if the model parameters are updated. Approaches to circumvent this issue can be devised and are the subject of ongoing research. We begin the analysis by providing the following definition.

Definition 3. (Optimal Data Compression): Given the selected parameterization and MPC formulation, an *optimal data compression* selects a dataset $\bar{\mathcal{D}} \subseteq \mathcal{D}$ such that

$$\left|\bar{\mathcal{D}}\right| = \min_{\hat{\mathcal{D}}} \ \left|\hat{\mathcal{D}}\right| \qquad \text{s.t. } \mathcal{S}_{\hat{\mathcal{D}}} \equiv \mathcal{S}_{\mathcal{D}}. \tag{34}$$

Hence, an optimal data compression retains the minimum amount of data required to represent the set $\mathcal{S}_{\mathcal{D}}$. In the following, we assume that A, B, and $\mathbf{b}$ are fixed and exploit the model to achieve an optimal data compression.

The introduction of the nominal model (20c) allows us to restructure the data and only store the noise $\mathcal{W} := \{ \mathbf{w}_0, \dots, \mathbf{w}_n \}$, obtained from

$$\mathbf{w}_k = \mathbf{s}_{k+1} - (A\mathbf{s}_k + B\mathbf{a}_k + \mathbf{b}). \tag{35}$$

By using (35), we rewrite the SDC as

$$\mathcal{S}_{\mathcal{D}} = \mathcal{S}_{\mathcal{W}} := \{\boldsymbol{\theta} \,|\, M\mathbf{w} \leq \mathbf{m} \,\forall\, \mathbf{w} \in \mathcal{W}\} \tag{36}$$

with $\boldsymbol{\theta}_{\mathbf{W}} = (M, \mathbf{m})$ a component of $\boldsymbol{\theta}$. By exploiting the model, we can, therefore, reduce the dimension of the space of the dataset from $2n_{\mathbf{s}} + n_{\mathbf{a}}$ to $n_{\mathbf{s}}$, since $(\mathbf{s}, \mathbf{a}, \mathbf{s}_+) \in \mathbb{R}^{n_{\mathbf{s}} \times n_{\mathbf{a}} \times n_{\mathbf{s}}}$ and $\mathbf{w} \in \mathbb{R}^{n_{\mathbf{s}}}$. Note that this is only possible if one neglects the dependence of $\mathbf{W}_{\boldsymbol{\theta}}$ (and therefore, $\mathcal{W}$) on $\mathbf{s}, \mathbf{a}$.

The dispersion set approximation related with (36) is then

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) = \{\, A\mathbf{s} + B\mathbf{a} + \mathbf{b} + \mathbf{w} \,|\, \forall\, \mathbf{w} \,\text{s.t.}\, M\mathbf{w} \leq \mathbf{m} \,\} \tag{37}$$

such that (25) and (27) used in constraint tightening are LPs. Additionally, it is cheap to: 1) evaluate if $\mathbf{s}_+ \in \hat{\mathbf{S}}_+$ by using (35) and 2) verifying that $\boldsymbol{\theta} \in \mathcal{S}_{\mathcal{W}}$, i.e., $M\mathbf{w}_k \leq \mathbf{m}$ holds, since both operations only require few matrix-vector operations. However, while 1) requires a single evaluation of the inequality in (37), 2) requires to evaluate the inequality for each data point in (36).

The use of the model and the convexity assumption make it possible to further compress data: Any point in the interior of the convex hull of set $\mathcal{W}$ does not provide any additional information regarding (36), such that the convex hull

$$\bar{\mathcal{W}} := \mathrm{Conv}(\mathcal{W})$$

carries all necessary information. Constructing the convex hull facet representation can be a rather expensive operation, which can in general not be done online. However, checking whether a sample $\mathbf{w}_k$ lies inside the convex hull $\bar{\mathcal{W}}$ of a set of samples $\mathcal{W}$ can be done without building the facet representation of the convex hull. To this end, we define the LP

$$\zeta := \min_{\mathbf{z}} \ \sum_{i=1}^{|\bar{\mathcal{W}}|} \mathbf{z}_i \qquad \text{s.t. } \hat{\mathbf{w}} = \sum_{i=1}^{|\bar{\mathcal{W}}|} \mathbf{z}_i \mathbf{w}_i, \qquad \mathbf{z} \geq 0 \tag{38}$$

and exploit the following result.

Proposition 3: Assume that $\mathbf{0} \in \mathrm{Conv}(\mathcal{W})$, then

$$\zeta \leq 1 \Leftrightarrow \hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W})$$

with ζ from (38).

Proof: The definition of convex hull implies $\hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W}) \Rightarrow \exists \quad \mathbf{z} \geq 0, \|\mathbf{z}\|_1 = 1 \quad \text{s.t.} \quad \hat{\mathbf{w}} = \sum_{i=1}^{|\bar{\mathcal{W}}|} z_i \mathbf{w}_i$, which proves $\zeta \leq 1 \Leftarrow \hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W})$. This also covers the implication $\zeta = 1 \Rightarrow \hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W})$. The implication $\zeta < 1 \Rightarrow \hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W})$ is proven by noting that $\zeta \mathbf{w}_i \in \mathrm{Conv}(\mathcal{W}) \quad \forall \quad \zeta \in [0, 1]$. The case $\zeta < 1$ is thus immediately reconducted to the case $\zeta = 1$. ■

Formulation (38) is very convenient, as it is always feasible, provided that $\mathcal{W}$ spans the full space $\mathbb{R}^{n_{\mathbf{w}}}$, which is a minimum reasonable requirement in this context, since it guarantees that every vector in $\mathbb{R}^{n_{\mathbf{w}}}$ can be obtained as a linear combination of $\mathbf{w}_i$. Note also that the assumption $\mathbf{0} \in \mathrm{Conv}(\mathcal{W})$ is a rather mild requirement on the accuracy of the model, as it always holds when standard system identification techniques are deployed to estimate the model parameters $A, B,$ and $\mathbf{b}$.

We prove the efficiency of the convex hull approach in the following theorem.

Theorem 1. (Convex Hull Optimality): Given the dispersion set $\hat{\mathbf{S}}_+$ defined in (37), the convex hull $\bar{\mathcal{W}}$ of the state transition noise $\mathcal{W}$ is an optimal data compression.

Proof: The definition of $\bar{\mathcal{W}}$ implies that any point in $\mathcal{W}$ can be obtained as the convex combination of points in $\bar{\mathcal{W}}$. By definition, we have that $\mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}(\mathbf{w}_1) \leq 0$, $\mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}(\mathbf{w}_2) \leq 0$, for all $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}$. For $\beta \in [0, 1]$, we define $\mathbf{w}_\beta := \beta \mathbf{w}_1 + (1 - \beta)\mathbf{w}_2$ such that

$$\mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}(\mathbf{w}_\beta) \leq \beta \mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}(\mathbf{w}_1) + (1 - \beta)\mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}(\mathbf{w}_2) \leq 0$$

since convexity of $\bar{\mathcal{W}}$ is equivalent to convexity of $\mathbf{g}_{\boldsymbol{\theta}}^{\mathbf{w}}$. This entails that $\mathbf{w}_\beta \in \mathcal{W}$, such that any set $\hat{\mathbf{S}}_+$ computed using $\bar{\mathcal{W}}$ satisfies

$$\mathbf{s}_{k+1} \in \hat{\mathbf{S}}_+(\mathbf{s}_k, \mathbf{u}_k, \boldsymbol{\theta}) \qquad \forall\, (\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{u}_k) \in \mathcal{D}$$

i.e., $\forall\, \mathbf{w} \in \mathcal{W}$. Finally, by removing any data point from $\bar{\mathcal{W}}$ one cannot guarantee that $\mathcal{S}_{\bar{\mathcal{W}}} = \mathcal{S}_{\mathcal{W}}$, such that $\bar{\mathcal{W}}$ solves (34). ■

In order to construct the convex hull $\bar{\mathcal{W}}$, once a new sample $\mathbf{w}_{\mathbf{k}}$ is available, we check whether $\mathbf{w}_k \in \bar{\mathcal{W}}$ by solving (38). In case $\zeta \leq 1$, $\bar{\mathcal{W}}$ does not need to be updated; otherwise, the new point is added to $\mathcal{W}$, such that $\bar{\mathcal{W}}$ is implicitly updated. Note that the LP (38) has linear complexity in the amount of vertices $|\bar{\mathcal{W}}|$.

Remark 6: As data are collected, $|\mathcal{D}|$ becomes indefinitely large. Even though $\bar{\mathcal{W}}$ is optimal, also $|\bar{\mathcal{W}}|$ can become indefinitely large, though arguably at a lower rate than $|\mathcal{D}|$. This is a fundamental issue of any sample-based SDC or set membership approach, unless additional assumptions are introduced. Simple strategies such as limiting the maximum amount of

vertices/facets to be used for an approximation $\hat{\mathcal{W}}$ of the convex hull could be devised, at the price of introducing conservatism. Such investigations are beyond the scope of this article and require future research.

Remark 7: One must take extra care if the model parameters A and B are updated. Indeed, if one updates $A, B,$ and $\mathbf{b}$ to $\tilde{A}, \tilde{B},$ and $\tilde{\mathbf{b}}$, then the new noise satisfies

$$\tilde{\mathbf{w}}_k = \mathbf{s}_{k+1} - \tilde{A}\mathbf{s}_k - \tilde{B}\mathbf{a}_k - \tilde{\mathbf{b}}$$

such that the noise update

$$\tilde{\mathbf{w}}_k - \mathbf{w}_k = ((A - \tilde{A})\mathbf{s}_k + (B - \tilde{B})\mathbf{a}_k + \mathbf{b} - \tilde{\mathbf{b}})$$

is state-action dependent for all $\tilde{A} \neq A$, $\tilde{B} \neq B$. Therefore, any change in those parameters requires one to recompute the noise vector $\mathbf{w}_k$ for all recorded state-action pairs. Updates in parameter $\mathbf{b}$ instead can be performed without much complication and simply entail a shift of the noise, which is state-action independent, such that $\tilde{\mathcal{W}} = \mathcal{W} + \mathbf{b} - \tilde{\mathbf{b}}$.

C. Further Observations on the Sample-Based SDC

The convex hull $\bar{\mathcal{W}}$ is the smallest set encompassing all observed samples, i.e., it is the smallest set satisfying the SDC (15). Therefore, it is optimal both in terms of volume and in terms of cost, i.e., in the sense of Definition 2. Hence, one might be tempted to select $\mathbf{W}_{\boldsymbol{\theta}} = \bar{\mathcal{W}}$ to reduce conservatism in MPC (20) as much as possible. However, $\bar{\mathcal{W}}$ is typically composed of a very large amount of facets, which renders the constraint tightening procedure very costly and results in a terminal constraint of high dimension. In practice, a set $\mathbf{W}_{\boldsymbol{\theta}}$ of fixed and low complexity is preferred, hence, the importance of enforcing set membership optimality as per Definition 2, i.e., through RL.

Thus far, we have not discussed how the set $\mathbf{W}_{\boldsymbol{\theta}}$ is represented. However, the choice of representation becomes important when dealing with large amounts of data. Moreover, the parameterization of $\mathbf{W}_{\boldsymbol{\theta}}$ should be selected consistently with the algorithm used for solving the robust MPC problem. Convex polytopes can be parameterized using the so-called facet or vertex representation. In case of facet representation, the set is parameterized as $\mathbf{W}_{\boldsymbol{\theta}} := \{\, \mathbf{w} \,|\, M\mathbf{w} \leq \mathbf{m} \,\}$ with parameter $\boldsymbol{\theta}_{\mathbf{W}} = (M, \mathbf{m})$. In case of vertex representation, the set is parameterized as $\mathbf{W}_{\boldsymbol{\theta}} := \{\, \mathbf{w} \,|\, \mathbf{w} \in \mathrm{Conv}(\{\mathbf{v}_0, \ldots, \mathbf{v}_m\}) \,\}$ with parameter $\boldsymbol{\theta}_{\mathbf{W}} = (\mathbf{v}_0, \ldots, \mathbf{v}_m)$, i.e., the vertices the polytope.

Differently from $\mathbf{W}_{\boldsymbol{\theta}}$, for the convex hull $\bar{\mathcal{W}}$, we use the vertex representation. This allows a simpler and less computationally demanding construction and incremental update of $\bar{\mathcal{W}}$. This advantage, however, results in a more costly evaluation of $\mathbf{w}_k \in \bar{\mathcal{W}}$ with respect to a facet representation. Finally, this makes it simple to enforce the SDC (36), which becomes

$$\mathcal{S}_{\bar{\mathcal{W}}} = \left\{ \boldsymbol{\theta} \,|\, M\mathbf{w} \leq \mathbf{m} \,\forall\, \mathbf{w} \in \bar{\mathcal{W}} \right\}. \tag{39}$$

The question on which representation is the most convenient for the convex hull $\bar{\mathcal{W}}$ is still open and will be further investigated in future research, which will also consider a combination of both the facet and vertex representation. We provide next some fundamental observations regarding $\bar{\mathcal{W}}$, its cardinality and its relationship with the nominal model.

Remark 8: As a consequence of Fundamental Limitation 1, in any sample-based context, it is possible that a new sample $\mathbf{w}_k$ falls out of the convex hull of previous samples, i.e., $\mathbf{w}_k \notin \bar{\mathcal{W}}$, such that potentially $M_i\mathbf{w}_k \geq \mathbf{m}_i$ for some i. In this case, one needs to instantaneously adapt the SDC (15). A straightforward adaptation of $\mathcal{S}_{\bar{\mathcal{W}}}$ is obtained as

$$\mathbf{m}_i \leftarrow \max(\mathbf{m}_i, M_i\mathbf{w}_k). \tag{40}$$

This enlargement of the uncertainty set $\mathbf{W}_{\boldsymbol{\theta}}$ entails an enlargement of the dispersion set, such that the constraints will be further tightened. This can jeopardize recursive feasibility of MPC (20) such that it is necessary to deploy the constraint relaxation proposed in Section III-C.

V. Safe RL MPC Implementation

After having introduced all necessary components of the RL-MPC scheme, in this section, we focus on the safe RL problem, discuss more in detail the RL problem and present some open research questions.

Most recursive RL approaches use only the current sample and update the parameter recursively as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha(\boldsymbol{\theta}^* - \boldsymbol{\theta}_k) \tag{41}$$

with $\boldsymbol{\theta}^* = \boldsymbol{\theta}_k + \nabla_{\boldsymbol{\theta}}\psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta})$ and ψ defined, e.g., as per (6) or (7).

This means that the update is the solution (or a step of stochastic gradient descent) of an unconstrained optimization problem. Since we need to enforce the SDC, the safe RL problem (18) yielding $\boldsymbol{\theta}^*$ is constrained. By relying on the developments of Section IV, we formulate (18) as

$$\min_{\boldsymbol{\theta}} \ \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \tag{42a}$$

$$\text{s.t. } H \succ 0, \qquad P \succ 0 \tag{42b}$$

$$M\mathbf{w} \leq \mathbf{m} \qquad \forall\, \mathbf{w} \in \bar{\mathcal{W}}. \tag{42c}$$

Problem (42a) (without constraints) can be seen as a standard RL formulation [20]. Positive-definite constraints (42b) are introduced to make sure that MPC is properly formulated and easily and efficiently solvable. The SDC (15) is imposed in (42c). If the model parameters A and B are updated, the SDC cannot be implemented as (15), but rather as $\boldsymbol{\theta} \in \mathcal{S}_{\mathcal{D}}$, with $\mathcal{S}_{\mathcal{D}}$ given by (33).

Remark 9: Any update in parameter $\boldsymbol{\theta}$ results in a modification of set $\mathbf{W}_{\boldsymbol{\theta}}$, such that the existence of a solution to the MPC problem could be jeopardized. Several options can be envisioned, including the following:

1) updating $\boldsymbol{\theta}$ only when feasible;
2) reducing the step size until feasibility is recovered;
3) enforcing feasibility as an additional constraint in (42).

Any of the strategies 1)–3) guarantees the feasibility of the updated robust MPC scheme since, provided that MPC is correctly formulated, by Proposition 1 initial feasibility entails recursive feasibility. In turn, this entails safety of the parameter update. It is worth mentioning that we did not encounter feasibility issues in the simulations we performed. Nevertheless, future research will investigate this issue in depth.

Remark 10: Both $Q_{\boldsymbol{\theta}}$ and $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ depend on the constraint tightening procedure, such that their first-order sensitivities can be discontinuous. Consequently, also the first-order sensitivities of ψ are nonsmooth. This is the case when SC does not hold either in the MPC problem (20) or in the constraint tightening problems (25) and (27), i.e., when some constraint(s) are weakly active, such that infinitesimal perturbations could cause an active-set change. In principle, this could create problems to algorithms for continuous optimization. However, the set on which SC does not hold has zero measure, such that the probability that one sample falls onto one of these points is zero, and the RL solution is unaffected.

Since the main concern of this article is safety, we present next how to guarantee safety also during exploration, i.e., when the action applied to the system is not given by (20). Afterwards, we will further discuss open research questions and possible ways of addressing them.

A. Safe Exploration

The proposed formulation guarantees safety during exploitation, i.e., whenever the policy is given by (21). However, specific care needs to be taken during exploration, i.e., when the optimal policy is perturbed, since also in this phase constraint satisfaction must not be jeopardized. Exploration is typically performed by picking a random action among all feasible actions with a given probability distribution. The main difficulty in enforcing safety is related to the complexity of the set of safe actions

$$\mathbf{A}(\mathbf{s}_0) := \{\, \mathbf{a}_0 \,|\, \exists\, \mathbf{a}_1, \dots \text{ s.t. } C\mathbf{s}_k + D\mathbf{a}_k + \bar{\mathbf{c}} \leq 0 \,\forall\, \mathbf{w} \in \bar{\mathcal{W}} \,\}.$$

Since this set is implicitly approximated within robust MPC, this issue can be tackled by using a modified version of the robust MPC problem (20)

$$\min_{\mathbf{x},\mathbf{u}} \quad (20a) + f\,(\mathbf{u}_0, \mathbf{q}) \tag{43a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{s}, \ (20c), (20d), (20e) \tag{43b}$$

with either $f(\mathbf{u}_0, \mathbf{q}) := \rho \|\mathbf{u}_0 - \mathbf{q}\|$, or $f(\mathbf{u}_0, \mathbf{q}) := \mathbf{q}^\top \mathbf{u}_0$, and a randomly chosen $\mathbf{q}$. Note that performing constrained exploration in policy gradient methods can produce a biased gradient estimation. Simple strategies to tackle this issue are the subject of ongoing research.

Theorem 2: Consider a convergent RL scheme solving problem (42), where robust MPC (20) is used as function approximator; available data are handled as detailed in Section IV; and exploration is performed according to (43). Assume that all new data yield $\mathbf{w}_k \in \bar{\mathcal{W}}$. Then, the RL scheme is safe in the sense of Definition 1, i.e., $\mathbb{P}[\exists\, k \,\text{s.t.}\, \boldsymbol{\xi}(\mathbf{s}_k, \mathbf{a}_k) > 0] \leq 1 - \eta(\mathcal{D})$. Additionally, the scheme is optimal in the sense of Definition 3.

Proof: We observe that $\eta(\mathcal{D})$ quantifies the probability that it is possible that a scenario is unaccounted for in the construction of $\hat{\mathbf{S}}_+$. Moreover, we observe that if $\hat{\mathbf{S}}_+$ does not account for all possible scenarios, then $\mathbf{w}_k \notin \bar{\mathcal{W}}$ can happen for some k. As a result, one can verify that

$$\mathbb{P}\left[\exists\, k \,\text{s.t.}\, \mathbf{w}_k \notin \bar{\mathcal{W}}\right] \leq 1 - \eta(\mathcal{D}).$$

Algorithm 1: RL MPC.

1: **if** Explore **then**
2: Get $\mathbf{a}$ from MPC (43)
3: **else**
4: Get $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s})$ from MPC (20)
5: Observe $\mathbf{s}, \mathbf{a} \to \mathbf{s}_+$
6: Compute $\mathbf{w}$, solve (38) to update $\bar{\mathcal{W}}$
7: Get $Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$, $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})$, $V_{\boldsymbol{\theta}}(\mathbf{s}_+)$
8: **if** Solve RL **then**
9: Compute RL step: e.g., by solving (42)
10: **if** Update $\boldsymbol{\theta}$ feasible **then**
11: Recompute constraint tightening (25), (27)

We note that, by Proposition 1, $\mathbf{w}_k \in \bar{\mathcal{W}}$ implies that robust MPC (20) is recursively feasible. Moreover, exploration is performed using (43), i.e., (20) with a modified cost, such that recursive feasibility is also preserved by Proposition 1. This entails that a constraint violation can only occur if $\mathbf{w}_k \notin \bar{\mathcal{W}}$. The converse, however, is not true in general, since there might exist $\mathbf{w}_k \notin \bar{\mathcal{W}}$, which does not entail a constraint violation. Consequently, the probability that the constraints be violated at any time is bounded from above by $1 - \eta(\mathcal{D})$ as

$$\mathbb{P}\left[\exists\, k \,\text{s.t.}\, \boldsymbol{\xi}(\mathbf{s}_k, \mathbf{a}_k) > 0\right] \leq \mathbb{P}\left[\exists\, k \,\text{s.t.}\, \mathbf{w}_k \notin \bar{\mathcal{W}}\right] \leq 1 - \eta(\mathcal{D}).$$

Data compression optimality (Definition 3) is a direct consequence of Theorem 1. ■

Remark 11: We ought to stress that assuming that RL converges to a minimum of $\mathbb{E}[\psi]$ is standard. However, assuming convergence to a local minimum of J can be a strong assumption, which is typically not met by Q-learning and SARSA, while actor-critic methods are typically expected to converge to a local minimum of J for the given parameterization. Optimality in the sense of Definition 2 (set membership optimality) could then be claimed for convergent actor-critic methods, while it is reasonable to expect a certain degree of suboptimality with Q-learning.

We provide an overview of the computations performed by RL-MPC in Algorithm 1. Note that we introduced lines 8 and 10 since the RL problem, and consequently, the recomputation of the constraint tightening and terminal set might in principle be updated less often than the feedback sampling time. In that case, a batch RL approach could be used instead of a recursive one and the real-time requirements would only concern MPC. Additionally, we stress that in our MATLAB implementation, which was only partially made efficient, the constraint tightening procedure on line 11 required approximately twice the computation time of MPC on lines 2 and 4. The computation of the RL step on line 9 was approximately 5 times faster.

B. Discussion on the Proposed Approach

We discuss next a few open research questions and comment on possible extensions to the proposed framework.

1) Safety: Our safety definition is based on the assumption that all future samples satisfy $\mathbf{w}_k \in \bar{\mathcal{W}}$. In practice this assumption, though standard in robust MPC and SMSI, can be rather

strong. However, this is a fundamental problem of safety: one can never guarantee *a priori* that a bounded set contains all possible future realizations of an unknown stochastic process. In Section III-C, we have proposed a simple and practical approach to retain MPC feasibility even in the case $\mathbf{w}_k \notin \bar{\mathcal{W}}$. However, with this approach safety is potentially lost every time a sample falls out of the convex hull $\bar{\mathcal{W}}$. Safety, however, is typically quickly recovered, as the RL parameter is instantaneously adjusted to account for the new sample. As also highlighted in Section II and Remark 8, this temporary loss of safety is a fundamental issue for any sample-based approach, unless additional assumptions are introduced. While one could envision the derivation of some measure of reliability of the identified set to be used to provide stronger guarantees, such investigation is beyond the scope of this article and will be the subject of future research.

2) Approximation Quality: It has been proven in [11, Th. 1] that, provided that the MPC parameterization is rich enough, the correct value and action-value functions can be recovered exactly. In the context of this article, however, the parameterization of the noise set is approximate by construction. Consider partitioning the parameter vector as $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\theta}_{\mathbf{W}})$, where $\boldsymbol{\theta}_{\mathrm{c}}$ is the parameter vector directly defining the cost and constraint functions, similarly to the parameterization of [11], while $\boldsymbol{\theta}_{\mathbf{W}}$ is the parameter vector parameterizing function $\mathbf{g}_{\boldsymbol{\theta}} := \mathbf{g}_{\boldsymbol{\theta}_{\mathbf{W}}}$. Then, provided that the parameterization of the cost and constraint functions through $\boldsymbol{\theta}_{\mathrm{c}}$ is rich enough, the value, action-value functions and policy can be recovered exactly on the feasible domain of the robust MPC (20). Therefore, as opposed to the result of [11], the equivalence only holds on the subset of the feasible domain of the RL problem that can be described by the chosen parameterization of $\mathbf{g}_{\boldsymbol{\theta}}$.

Since the hypothesis of a perfect parameterization is unrealistic in most relevant applications, Q learning and SARSA will in general not deliver the best performance that can be attained with the selected parameterization, since these algorithms aim at fitting the action-value function rather than directly optimizing performance. It is, therefore, appealing to resort to policy gradient approaches, which seek the direct minimization of J by manipulating $\boldsymbol{\theta}$. While in principle, the proposed approach is well suited for policy gradient methods (both stochastic and deterministic), the main difficulty that needs extra care to be handled is related to the exploration strategy to be deployed, which in general introduces a bias in the gradients, such that convergence to a local optimum is hindered.

3) Model Adaptation: In principle, one could choose to let RL update any of the parameters (22) of the MPC scheme (20), including the model parameters A, B, and $\mathbf{b}$. The model used by MPC plays an important role in the following:

1) the definition of the value and action-value function;
2) the conservatism of the uncertainty set approximation $\mathbf{W}_{\boldsymbol{\theta}}$.

As observed in Remark 7, letting RL adapt the model parameters results in a large increase of computations. The feedback matrix K, however, can be adapted by RL without major issues. Since it has an impact on the size of the uncertainty propagation, and through $\mathbf{c}_k$ and $\mathbf{g}$, on the cost, letting RL adapt K is expected to further reduce the closed-loop cost by reducing the tightening of the specific constraint components $\mathbf{c}_{k,i}$, $\mathbf{g}_j$ corresponding to constraints, which are active in the execution of the the control task. Note that selecting a K that is optimal for the given task is an open issue that can be addressed in a data-driven fashion using the approach we propose.

In [11, Th. 1] and [20, Cor. 2], it is proven that RL can find $Q_{\boldsymbol{\theta}} = Q$, $V_{\boldsymbol{\theta}} = V$ and $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}$ even without adjusting the model, provided that the parameterization is descriptive enough. However, typically the parameterization is low-dimensional and cannot be expected to fulfill this requirement. Therefore, the question on whether adapting the model provides an advantage and on how to best adapt it is still open.

One possibility to improve the approximation quality could be to carry a fixed model parameterization A_0, B_0, $\mathbf{b}_0$ to be used for safety and one or several (possibly nonlinear) models $\mathbf{f}_{i,\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{u}_i)$, each associated with its cost to construct a more accurate prediction of the future cost distribution in a scenario-tree fashion. The investigation of alternative formulations will be the subject of future research.

4) Stability: The results on robust MPC are stronger than the one provided in Proposition 1: Asymptotic stability is proven, under the assumptions of having $\gamma = 1$ and a positive-definite stage cost, usually called *tracking* cost, as opposed to *economic* cost. In case of a tracking cost, the presence of a discount factor in the MPC formulation complicates the stability analysis. While one could foresee formulating (20) with $\gamma = 1$, even though $\gamma < 1$ in the RL problem, it is not immediately clear to which extent this inconsistency will impact on the ability to learn the correct policy. While providing approximation guarantees for Q-learning is arguably nontrivial, policy gradient methods will at least provide the best policy for the selected parameterization. In this context that would be the best policy with stability guarantees. By exploiting the results of [34], it should be possible to prove that stabilizing linear control laws can be recovered exactly for linear systems. However, a thorough investigation of this topic is the subject of ongoing research.

The distinction between tracking and economic cost is relevant in relation to the stability properties of the closed-loop system: A thorough discussion on economic MPC can be found in [35]–[37]. The linear-quadratic case, which is particularly relevant for our framework, has been analyzed in, e.g., [34] and [38]–[41]. We only recall here that all the conclusions drawn in [11] apply directly to our setup, i.e., the economic case can be reduced to the tracking case by additionally learning an initial cost.

VI. Simulation Results

We test our approach with two examples in simulations. First, we consider a linear system to easily interpret the results. Then, we consider a realistic example from the chemical industry: A nonlinear evaporation process.

A. Linear System

We consider a linear system having two states, such that we can easily visualize the behavior of RL-MPC. Consider a simple

linear system with dynamics and stage cost

$$\mathbf{s}_+ = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \mathbf{s} + \begin{bmatrix} 0.05 \\ 0.1 \end{bmatrix} a + \mathbf{w}$$

$$\ell(\mathbf{s}, \mathbf{a}) = \begin{bmatrix} \mathbf{s} - \mathbf{s}^{\mathrm{r}} \\ a - a^{\mathrm{r}} \end{bmatrix}^{\top} \operatorname{diag}\left(\begin{bmatrix} 1 \\ 0.01 \\ 0.01 \end{bmatrix} \right) \begin{bmatrix} \mathbf{s} - \mathbf{s}^{\mathrm{r}} \\ a - a^{\mathrm{r}} \end{bmatrix}$$

where $\mathbf{s} = (p, v)$. We formulate a problem with prediction horizon $N = 20$ and introduce the state and control constraints $-\mathbf{1} \leq \mathbf{s} \leq \mathbf{1}$ and $-10 \leq a \leq 10$. The real noise set is selected as a regular octagon. In order to illustrate the ability of RL to adapt the approximation of the noise set, we select to parameterize $\mathbf{W}_{\boldsymbol{\theta}}$ as a polytope with four facets. We compute the terminal cost matrix P and feedback gain K using the LQR formulation resulting from the nominal model and stage cost. The terminal feedback K is used for constraint tightening, as per Section III.

We simulate the RL-MPC scheme over 200 time steps in a scenario without exploration in which the reference is

$$p^{\mathrm{r}}(t) = \begin{cases} 1 & 25 \leq t \leq 120 \\ -1 & \text{otherwise} \end{cases}$$

$$v^{\mathrm{r}}(t) = 0, \qquad a^{\mathrm{r}}(t) = 0.$$

Since the setpoint reference is moving, for simplicity, we impose the terminal constraint centered around the origin. While this does not affect recursive feasibility, practical stability is harder to prove in this case. A thorough analysis of this aspect goes beyond the scope of this article, and we simply recall that such a terminal constraint induces a leaving arc in the MPC optimal control problem. This situation has been analyzed in [39], [40], and [42] in the context of economic MPC, concluding that under suitable assumptions verified by the system considered here, practical stability is obtained.

We update $\boldsymbol{\theta} = (M, \mathbf{m})$ according to (41) and (42) with $\alpha = 0.1$, ψ given by (6) (Q learning). Problem (42) is solved to full convergence at each step such that globalization techniques guarantee that $\boldsymbol{\theta}^*$ is a better fit than $\boldsymbol{\theta}_k$ for the sample at hand; positive-definiteness of the cost yields a well-posed MPC formulation; and the choice of parameter α is directly related to the horizon of a moving-average approximation of the expected value.

The simulation results are displayed in Figs. 2 and 3 in the form of snapshots comparing a representation of the solver at two different time instants. In particular, the constraint, RPI, and terminal sets are represented together with the trajectories and constraint tightening. Associated with these quantities, we also display the uncertainty set with the drawn samples, their convex hull and the approximation that RL learned. Initially, RL adapts the rather conservative approximation, therefore, enlarging the terminal set by reducing the required constraint tightening. When the setpoint moves, there is initially no gain in modifying the set adaptation, until at $k = 34$ the tightened constraint $p \leq 1$ becomes active. At this point, RL starts adapting the set approximation to better capture the shape of the uncertainty set in its top-right part, which is opposite to the bottom-left corner, which was better approximated before. Consequently, the terminal set is shifted toward the reference and

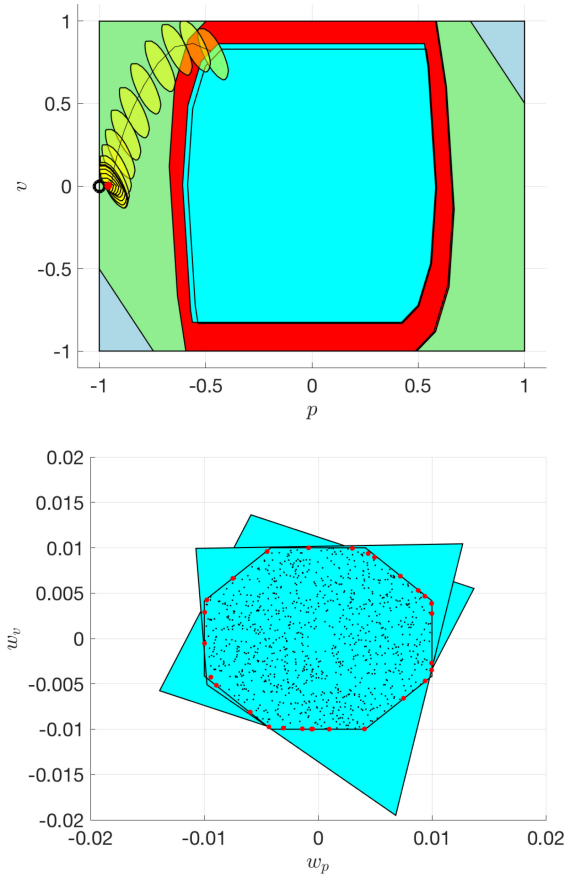

Fig. 2. Snapshots at $k = 0$ and $k = 24$. (Top) State space: State constraint set (light blue), state-input constraints using feedback matrix K (green), RPI set (red), and terminal constraint set $\mathcal{X}_{\mathrm{f}}$ (cyan for $k = 24$ and transparent for $k = 0$). Predicted trajectory at $k = 24$: Initial state (red dot), predicted trajectory (solid black line), reference $\mathbf{s}^{\mathrm{r}}$ (black circle), uncertainty tube (yellow). (Bottom) Noise space: True uncertainty set (transparent octagon), noise samples (black dots), vertices of their convex hull (red dots), and uncertainty set approximations $\mathbf{W}_{\boldsymbol{\theta}}$ (cyan sets, with $k = 0$ in the background). A better approximation $\mathbf{W}_{\boldsymbol{\theta}}$ ($k = 24$) enlarges $\mathcal{X}_{\mathrm{f}}$.

the $p \leq 1$ is tightened less, with an infinite-horizon difference of approximately 0.019. Since the tightening steady-state is quickly reached, this difference is visible in Fig. 3.

Note that $\mathbf{m}$ does in principle not need to be adjusted, as any adjustment in $\mathbf{m}$ can be equivalently obtained by suitably rescaling M. We performed the same simulations with $\theta = M$ and obtained qualitatively equivalent results, the detailed presentation of which we therefore omit. Finally, note that the convex hull of $\mathcal{W}$ is a polytope with 28 facets, as opposed to the used approximation $\mathbf{W}_{\boldsymbol{\theta}}$, which only has four facets.

We ran an additional simulation in which we let $\boldsymbol{\theta} = (M, K)$, such that RL also adapts the feedback matrix K used for constraint tightening. We remark that the problem of designing the terminal feedback and corresponding set $\mathcal{X}_\infty$ is nontrivial and many approaches have been developed. However, to the best of the authors' knowledge, none of these approaches explicitly accounts for the specific control task to be executed and rather aim at minimizing constraint tightening or maximizing the size of the RPI set. The simulation results are similar to the previous

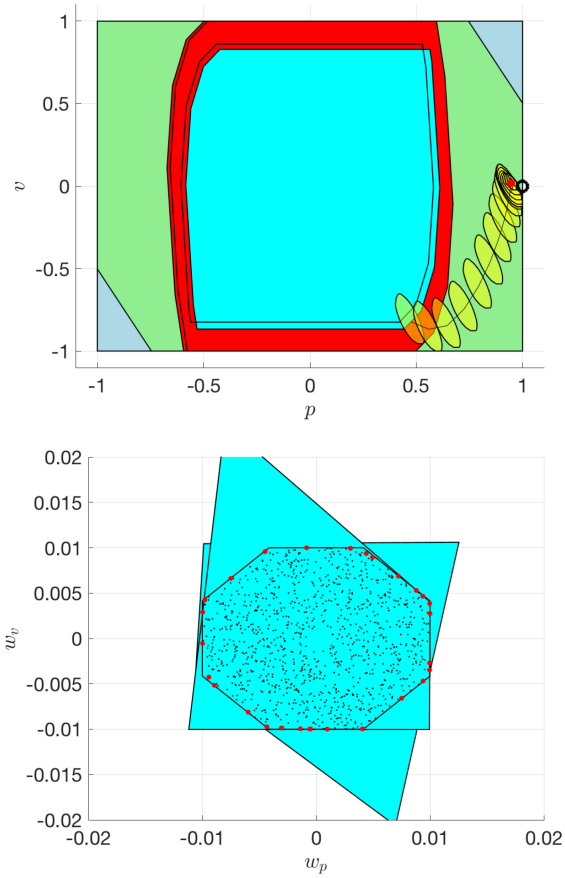

Fig. 3. Snapshots at $k = 34$ and $k = 109$: Same convention as Fig. 2, with predicted trajectory at $k = 109$. Both the RPI and terminal sets moved closer to the setpoint by a better approximation $\mathbf{W}_{\boldsymbol{\theta}}$ for the specific control task (see the bottom plot). Moreover, next to s^{r} the constraints are tightened more at $k = 36$ than afterwards.

case. However, RL acts on K to enlarge the RPI and terminal set while reducing the required constraint tightening, therefore, obtaining an increase in closed-loop performance.

B. Evaporation Process

Consider the evaporation process modeled in [43] and [44] and used in [38] and [45] to demonstrate the potential of economic MPC in the nominal case. The model has states $\mathbf{x} = (X_2,\ P_2)$ (concentration and pressure); controls $\mathbf{u} = (P_{100},\ F_{200})$ (pressure and flow); and dynamics given by [45]

$$M\dot{X}_2 = F_1X_1 - F_2X_2, \qquad C\dot{P}_2 = F_4 - F_5 \tag{44}$$

where $T_2 = aP_2 + bX_2 + c$, $T_3 = dP_2 + e$, $\lambda F_4 = Q_{100} - F_1C_{\mathrm{p}}(T_2 - T_1)$, $T_{100} = fP_{100} + g$, $Q_{100} = UA_1(T_{100} - T_2)$, $UA_1 = h(F_1 + F_3)$, $Q_{100} = UA_1(T_{100} - T_2)$, $UA_1 = h(F_1 + F_3)$, $Q_{200} = \frac{UA_2(T_3 - T_{200})}{1 + UA_2/(2C_{\mathrm{p}}F_{200})}$, $F_{100} = \frac{Q_{100}}{\lambda_{\mathrm{s}}}$, $\lambda F_5 = Q_{200}$, and $F_2 = F_1 - F_4$. The model parameters are given in [38].

Concentration X_1, flow F_2, and temperatures T_1, T_{200} are stochastic. In this example, we model them as a uniform distribution with $X_1 = 5 \pm 0.5$, $F_1 = 10 \pm 0.5$, $T_1 = 40 \pm 4$, $T_{200} = 25 \pm 5$. Additionally, the controller must satisfy bounds $(25, 40) \leq (X_2, P_2) \leq (100, 80)$ on the states and $100 \leq (P_{100}, F_{200}) \leq 400$ on the controls. The stage cost is given by

$$\ell(\mathbf{x}, \mathbf{u}) = \ 10.09(F_2 + F_3) + 600F_{100} + 0.6F_{200}$$

which entails that the nominal model is optimally operated at the steady state $\mathbf{x}_{\mathrm{s}} = (25, 49.74)$, $\mathbf{u}_{\mathrm{s}} = (191.71, 215.89)$, with stage cost $\ell(\mathbf{x}_{\mathrm{s}}, \mathbf{u}_{\mathrm{s}}) = \ell_{\mathrm{s}}$.

The system is linearized at $\mathbf{x}_{\mathrm{s}}, \mathbf{u}_{\mathrm{s}}$ to obtain a linear nominal model. The terminal set is centered at $\mathbf{x}_{\mathrm{c}} = (29, 53.57)$, and $\mathbf{u}_{\mathrm{c}} = (223.76, 221.61)$, which is a steady state for the linearized

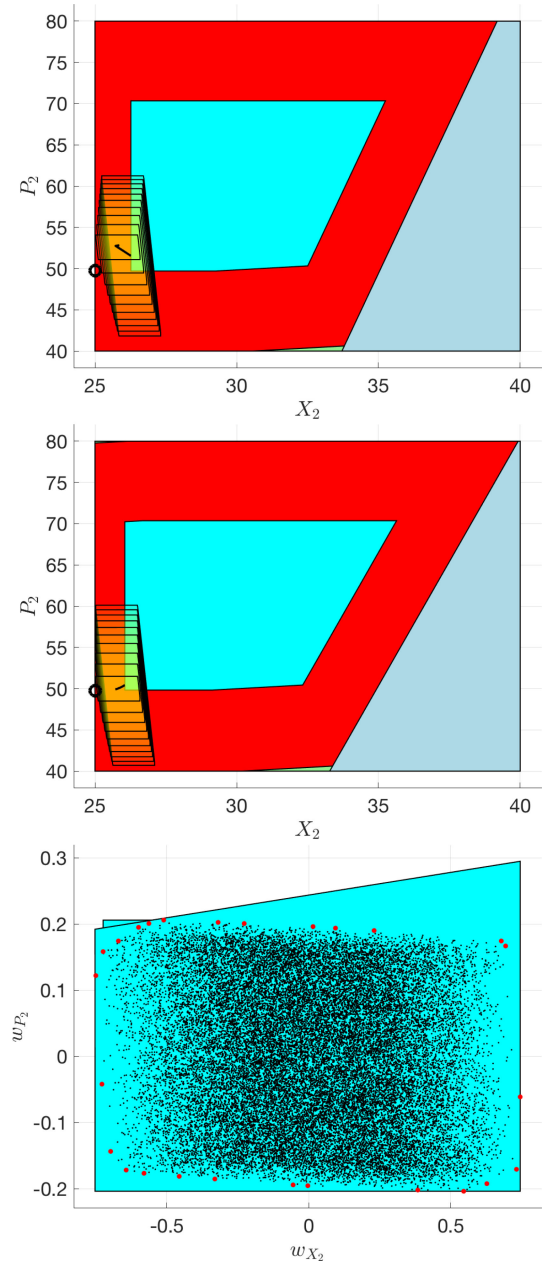

Fig. 4. Evaporation process. (Top and middle) RPI and terminal set at the beginning and end of the learning process, same color convention as Fig. 2. (Bottom) The uncertainty set approximation $\mathbf{W}_{\boldsymbol{\theta}}$.

dynamics. Since the safety constraints are already linear, we introduce discount factor $\gamma = 0.99$ and formulate a robust linear MPC problem of the form (20). To that end, we use the linearization at x_s and u_s, the quadratic cost obtained by applying the tuning procedure proposed in [38]. With the given stage cost and linear model, we obtain the LQR feedback K, which we use to stabilize the model error in computing the constraint tightening (25) and the terminal set (27). We then use RL to adjust parameter $\boldsymbol{\theta} = (\mathbf{h}, \mathbf{p}, M, K)$, i.e., the cost gradient, the uncertainty set, and the feedback matrix K. Since we formulate the robust MPC problem (20) using a positive-definite cost, while the RL cost ℓ is economic, we also learn a quadratic initial cost, as briefly discussed in Section V-B and fully justified in [11].

We let the Q-learning algorithm run for 10^4 samples with $\alpha = 10^{-2}$. We note that the parameters are no longer adjusted toward the end of the learning procedure, indicating convergence of the algorithm. The RPI and terminal set, as well as the uncertainty set approximation $\mathbf{W}_{\boldsymbol{\theta}}$ are displayed in Fig. 4 at the beginning and at the end of the learning process. One can note that $\mathbf{W}_{\boldsymbol{\theta}}$ becomes larger in order to better approximate the top left part of the uncertainty set. In combination with the adjustment of the feedback matrix K, this allows to shift the terminal set toward the reference, as shown in the top plot.

VII. Conclusion and Future Work

In this article, we have presented an RL algorithm, which is guaranteed to be safe in the sense of strictly satisfying a set of prescribed constraints given the available data. We have discussed both an innovative function approximation based on robust MPC and an efficient management of data that makes it possible to deal with very large amounts of data in real time. While linear MPC can be successfully applied also to nonlinear systems, as demonstrated with the second example, in case of strong nonlinearities, linear MPC might fail at providing satisfactory performance and even safety.

The proposed framework paves the road for several extensions, which are as follows:

1) one can easily foresee the use of robust MPC based on scenario trees (which is also suitable for nonlinear models);
2) the use of stochastic or deterministic policy gradient is expected to further improve performance, as discussed in Section II;
3) a formulation using the computational geometry approach for robustness and scenario trees for a refined cost approximation could be envisioned.

These research directions are the subject of ongoing research.

References

[1] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[2] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[3] S. Wang, W. Chaovalitwongse, and R. Babuska, "Machine learning algorithms in bipedal robot control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 728–743, Sep. 2012.

[4] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA, USA: MIT Press, 2007.

[5] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. 12th Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 1057–1063.

[6] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. I-387–I-395.

[7] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Univ. Cambridge, Cambridge, U.K., 1989.

[8] J. F. J. Garcia, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, pp. 1437–1480, 2013.

[9] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," 2018. [Online]. Available: https://arxiv.org/abs/1801.08757

[10] T. Pham, G. De Magistris, and R. Tachibana, "Optlayer—Practical constrained optimization for deep reinforcement learning in the real world," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 6236–6243.

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

[12] S. Gros, M. Zanon, and A. Bemporad, "Safe reinforcement learning via projection on a safe set: How to achieve optimality?" in *Proc. 21st IFAC World Congr.*, 2020.

[13] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," 2018, *arXiv:1803.08287*.

[14] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[15] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016.

[16] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe Model-based Reinforcement Learning with Stability Guarantees," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2017, pp. 908–918.

[17] R. Murray and M. Palladino, "A model for system uncertainty in reinforcement learning," *Syst. Control Lett.*, vol. 122, pp. 24–31, 2018.

[18] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Jul.–Sep. 2009.

[19] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[20] M. Zanon, S. Gros, and A. Bemporad, "Practical reinforcement learning of stabilizing economic MPC," in *Proc. Eur. Control Conf.*, 2019, pp. 2258–2263.

[21] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable MPC for end-to-end planning and control," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (NIPS'18). Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 8299–8310.

[22] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 5582–5588.

[23] J. B. Rawlings and D. Q. Mayne, "Postface to model predictive control: Theory and design," in, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill, 2012.

[24] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 2001.

[25] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[26] D. Mayne, "Robust and stochastic MPC: Are we going in the right direction?" *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 1–8, 2015.

[27] I. Kolmanovsky and E. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Math. Probl. Eng.*, vol. 4, no. 4, pp. 317–367, 1998.

[28] C. Büskens and H. Maurer, "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems," in *Online Optimization of Large Scale Systems*. Berlin, Germany: Springer, 2001, pp. 3–16.

[29] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering), 2nd ed. New York, NY, USA: Springer, 2006.
[30] P. Scokaert and J. Rawlings, "Feasibility issues in linear model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1649–1659, 1999.
[31] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester, U.K.: Wiley, 1987.
[32] H. Bock, "Recent advances in parameter identification techniques for ODE," in *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, P. Deuflhard and E. Hairer, Eds. Boston, MA, USA: Birkhäuser, 1983, pp. 95–121.
[33] D. Bertsekas and I. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Autom. Control*, vol. AC-16, no. 2, pp. 117–128, Apr. 1971.
[34] M. Zanon, S. Gros, and M. Diehl, "Indefinite linear MPC and approximated economic MPC for nonlinear systems," *J. Process. Control*, vol. 24, pp. 1273–1281, 2014.
[35] M. Diehl, R. Amrit, and J. Rawlings, "A Lyapunov function for economic optimizing model predictive control," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 703–707, Mar. 2011.
[36] R. Amrit, J. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annu. Rev. Control*, vol. 35, pp. 178–186, 2011.
[37] M. A. Müller, D. Angeli, and F. Allgöwer, "On necessity and robustness of dissipativity in economic model predictive control," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1671–1676, Jun. 2015.
[38] M. Zanon, S. Gros, and M. Diehl, "A tracking MPC formulation that is locally equivalent to economic MPC," *J. Process Control*, vol. 45, pp. 30–42, 2016.
[39] M. Zanon and T. Faulwasser, "Economic MPC without terminal constraints: Gradient-correcting end penalties enforce asymptotic stability," *J. Process Control*, vol. 63, pp. 1–14, 2018.
[40] T. Faulwasser and M. Zanon, "Asymptotic stability of economic NMPC: The importance of adjoints," in *Proc. IFAC Nonlinear Model Predictive Control Conf.*, 2018, pp. 157–168.
[41] L. Grüne and R. Guglielmi, "Turnpike properties and strict dissipativity for discrete time linear quadratic optimal control problems," *SIAM J. Control Optim.*, vol. 56, no. 2, pp. 1282–1302, 2018.
[42] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, pp. 725–734, 2013.
[43] F. Y. Wang and I. T. Cameron, "Control studies on a model evaporation process – constrained state driving with conventional and higher relative degree systems," *J. Process Control*, vol. 4, pp. 59–75, 1994.
[44] C. Sonntag, O. Stursberg, and S. Engell, "Dynamic optimization of an industrial evaporator using graph search with embedded nonlinear programming," in *Proc. 2nd IFAC Conf. Analysis Design Hybrid Syst.*, 2006, pp. 211–216.
[45] R. Amrit, J. B. Rawlings, and L. T. Biegler, "Optimizing process economics online using model predictive control," *Comput. Chem. Eng.*, vol. 58, pp. 334–343, 2013.

Sébastien Gros received the Ph.D. degree from École polytechnique fédérale de Lausanne, Lausanne, Switzerland, in 2007.

After a journey by bicycle from Switzerland to the Everest base camp in full autonomy, he joined a R&D group hosted at Strathclyde University focusing on wind turbine control. In 2011, he joined the KU Leuven, where his main research focus was on optimal control and fast MPC for complex mechanical systems. He joined the Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, in 2013, where he became an Associate Professor in 2017. He is currently a Full Professor with the Norwegian University of Science and Technology, Trondheim, Norway and a Guest Professor with the Chalmers University of Technology. His main research interests include numerical methods, real-time optimal control, reinforcement learning, and the optimal control of energy-related applications.

Mario Zanon (Member, IEEE) received the master's degree in mechatronics from the University of Trento, Trento, Italy, in 2010, the Diplôme d'Ingénieur degree from the Ecole Centrale Paris, Châtenay-Malabry, France, in 2010, and the Ph.D. degree in electrical engineering from the KU Leuven, Leuven, Belgium, in 2015.

He had research stays with the KU Leuven, University of Bayreuth, Chalmers University, and the University of Freiburg. He also had a Postdoctoral Researcher position with Chalmers University until the end of 2017 and is currently an Assistant Professor with the IMT School for Advanced Studies Lucca, Lucca, Italy. His research interests include numerical methods for optimization, economic model predictive control, optimal control, and estimation of nonlinear dynamic systems, in particular, for aerospace and automotive applications.