Safe Reinforcement Learning Using Robust MPC

Mario Zanon and Sébastien 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 partially neglected by the RL community, are central to the control community which 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 paper, we propose to combine RL and MPC in order to exploit the advantages of both and, therefore, obtain a controller which is optimal and safe. We illustrate the results with a numerical example in simulations.**

***Index Terms*—Reinforcement Learning, Robust Model Predictive Control**

I. Introduction

Reinforcement Learning (RL) is a sample-based technique for optimizing Markov Decision Processes (MDP) [1]. Instead of relying on a model of the probability distributions underlying the state transitions, observed state transitions and costs (or rewards) are the only input to the algorithm. The most influential success stories of RL include 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 $\boldsymbol{\pi}(\mathbf{s})$ for the real system either directly (policy search methods) [5], [6] or indirectly (SARSA, Q-learning) [7]. In the latter, the optimal policy $\boldsymbol{\pi}(\mathbf{s})$ is then 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 $\boldsymbol{\pi}$ or Q are typically approximated by a function approximator, usually a Deep Neural Network (DNN).

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 DNN as a function approximator. Some approaches have been developed in order to guarantee some form of safety: see, e.g., the excellent survey [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.

The combination of learning and control techniques has been proposed in, e.g., [9], [10], [11], [12], [13]. Some attempts at combining RL and the linear quadratic regulator have been presented in [14], [15]. To the best of our knowledge, [16], [17] are the first works proposing to use NMPC as a function approximator in RL. Strategies for providing some form of safety have been developed, see, e.g., [8] and references therein. However, 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 [16] and some of the approaches in [8]. Finally, no approach providing some form of stability guarantees has been developed yet.

In this paper we propose a formulation which addresses the issue of safety. We summarize next two existing approaches and the scheme we propose, which combines them:

a) MPC, Business-as-usual: In robust MPC approaches one first identifies a low-dimensional, and therefore computationally tractable, uncertainty set enclosing all data points using system identification techniques and then formulates a robust MPC problem as in (17).

b) RL, Business-as-usual: In RL, one typically lets the optimization procedure yield a policy which does not violate the constraints by penalizing violations with a suitably high cost. Safety is typically not guaranteed and only few results provide weak guarantees.

c) Safe RL-MPC: In the approach we propose, a robust MPC problem is formulated similarly to (a). However, similarly to (b), RL updates the parametrization of the robust MPC scheme, such that the safety constraint is updated online to reduce conservativeness.

Safe RL-MPC is based on the approach first advocated in [16], [17]. The scheme can be seen from two alternative points of view: (a) MPC is used as a function approximator within RL in order to provide safety and stability guarantees; and (b) RL is used in order to tune the MPC parameters and, therefore, improve 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 paper 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) model allows one to significantly reduce the amount of data to be stored. Further efficiencies are obtained by exploiting convexity and relying on a low-dimensional approximation of the uncertainty set.

Finally, traditional RL approaches are revised and adaptations are proposed in order to make them applicable to the proposed 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 [17]. While the deployment of Q-learning does not present difficulties, the case of actor-critic

M. Zanon is with the IMT School for Advanced Studies Lucca, Italy.
S. Gros is with the Department of Cybernetic, NTNU, Norway.

techniques requires some adaptation when the input space is continuous and restricted by safety requirements, as discussed in [18], [19].

The paper is structured as follows. We introduce the problem of safe RL in general terms in Section II. 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. Conclusions and an outline for future research are given in Section VII.

Notation: We use the following convention: a is scalar, $\mathbf{a}$ is a vector with components $\mathbf{a}_i$, A is a matrix with rows A_i and $\mathbf{A}$, $\mathcal{A}$ 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 exception will be functions J, V, Q which are scalar, but usually denoted by capital letters in the literature.

II. Safe Data-Based NMPC Tuning

In this paper, we consider real system dynamics described as a Markov Process (MP), with state transitions having the underlying conditional probability density

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

denoting the probability density of state transition $\mathbf{s}, \mathbf{a} \to \mathbf{s}_+$. We furthermore consider a deterministic policy delivering the control input as $\mathbf{a} = \boldsymbol{\pi}\left(\mathbf{s}\right)$, 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 instantaneous reward in RL and γ si a discount factor, typically selected smaller than 1 in RL and equal to 1 in MPC. 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 $V_\star$, $Q_\star$, $\boldsymbol{\pi}_\star$ in order to find parameter $\boldsymbol{\theta}^\star$ which (approximately) solves (2) either directly or by approximately solving (4) in a sampled-based fashion. For both approaches we summarize this as

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

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

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

in Q learning; 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. In this paper 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 $\operatorname{supp}\left(\tau^{\hat{\boldsymbol{\pi}}}\right)$ the support of the distribution of the MP (1) subject to policy $\hat{\boldsymbol{\pi}}$. 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 process (1) has finite support.

Enforcing (8) poses two major challenges: (i) either the support of (1) or the support of $\tau^{\hat{\boldsymbol{\pi}}}$ must be known or estimated; (ii) given knowledge on either supports, a policy satisfying (8) must be designed.

Problem (i) 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. If collected data is informative, in the limit for an infinite amount of data the support is expected to be reconstructed exactly. A theoretical justification of this is beyond the scope of this paper. In order to construct an approximation of the MP support, we first define the dispersion set confining the state transitions as:

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

Similarly to the standard RL case, also for $\mathbf{S}_+$ we introduce a function approximation based on parameter $\boldsymbol{\theta}$:

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

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.

Because full information on the state transitions is not available, enforcing constraint (11) for all $\mathbf{s}, \mathbf{a}$ is impossible. We therefore need to resort to the sample-based form of (11)

$$\mathbf{s}_{k+1} \in \hat{\mathbf{S}}_+(\mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}), \qquad \forall\ k. \tag{12}$$

If the parametrization of $\hat{\mathbf{S}}_+$ is not rich enough, the SDC (11) cannot be satisfied tightly, i.e., $\hat{\mathbf{S}}_+ \neq \mathbf{S}_+$, such that conservativeness is typically introduced. Consequently, this outer approximation should be selected such that its detrimental impact on the closed-loop performance of the policy is minimized.

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

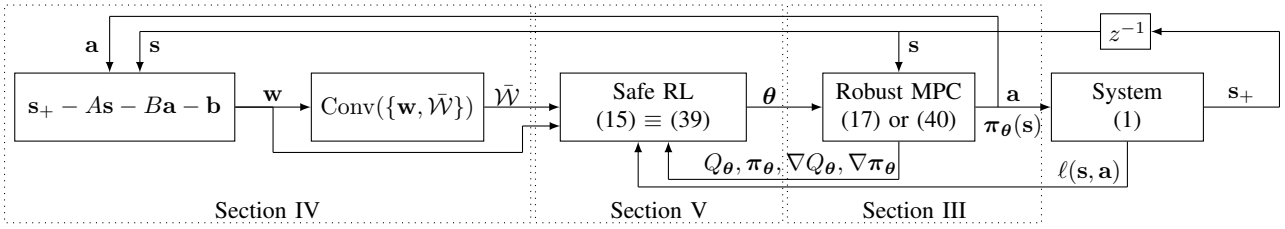

Fig. 1: Schematics of the proposed setup: data are used to construct the SDC based on $\bar{\mathcal{W}}$ and to evaluate the cost in (5). This cost depends on $Q_{\boldsymbol{\theta}}, V_{\boldsymbol{\theta}}$ obtained from MPC, and ℓ. MPC controls the system. The signal toggling between exploitation and exploration is omitted to avoid confusion, and is a signal sent from RL to switch between MPC (17) and (40).

Providing such 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 [20], [21], [22]. 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 [16], [17]. It has been proven in [16] 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 parametrization is rich enough.

In order to guarantee safety, robust MPC relies on the propagation of the dispersion set in time, defined as

$$\mathbf{S}^{\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}}_{k+1} := \hat{\mathbf{S}}_+ \left(\mathbf{S}^{\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}}_{k}, \boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}} \left(\mathbf{S}^{\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}}_{k} \right), \boldsymbol{\theta} \right), \tag{13}$$

where policy $\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}$ is introduced to stabilize the dispersion set and, ideally, one would select $\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}} = \boldsymbol{\pi}_{\boldsymbol{\theta}}$. Unfortunately, this poses severe computational challenges in the context of MPC and an auxiliary policy (typically affine) is preferred in order to recover a computationally tractable formulation [21]. More details on this topic will be given in Section III.

Based on the time propagation of the dispersion set, we provide the following definition of safety.

Definition 1 (Safe Policy): A policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is labelled as safe if it satisfies

$$\boldsymbol{\xi}(\mathbf{S}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_k, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{S}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_k)) \leq 0, \qquad \forall\, k \geq 0. \tag{14}$$

By construction, robust MPC delivers a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ which satisfies constraints $\boldsymbol{\xi}$ at all future times, provided that the SDC (12) holds for all k. Even though the dispersion set propagation $\mathbf{S}^{\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}}_k$ is computed based on $\boldsymbol{\pi}^{\mathrm{s}}_{\boldsymbol{\theta}}$, the constraints are guaranteed to hold also for $\mathbf{S}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_k$.

The safe RL problem is formulated, similarly to (5), as

$$\boldsymbol{\theta}^\star := \min_{\boldsymbol{\theta}} \quad \sum_{k=0}^{n} \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}) \tag{15a}$$

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

As stressed above, if the SDC holds, MPC delivers a safe policy by construction, such that safety is fully demanded to the SDC enforcement. Though in principle the SDC needs to be enforced for each sample, in Section IV we will propose an approach to largely reduce the amount of constraints.

Remark 1: The RL Problem (15) is typically solved by derivative-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.

In the scheme we propose, at each time instant, the safe RL framework performs the following steps: (a) MPC is solved and differentiated; (b) the optimal input is applied to the system; (c) state transitions are observed and data is collected to form the sample-based SDC (12); (d) the RL problem (15) is solved and parameter $\boldsymbol{\theta}$ is updated. A scheme of the setup is displayed in Figure 1, with reference to the section where the specific component is presented.

III. Robust MPC Based on Invariant Sets

In this section we assume safety constraints defined by an affine function $\mathbf{g}_{\boldsymbol{\theta}}$, i.e.,

$$C\mathbf{s} + D\mathbf{a} + \bar{\mathbf{c}} \leq \mathbf{0}, \tag{16}$$

such that $\hat{\mathbf{S}}_+$ is a polytope. We formulate a function approximator based on tube-based robust linear MPC:

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) := \\ \min_{\mathbf{z}} \quad & \sum_{k=0}^{N-1} \gamma^k \left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top H \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \mathbf{h}^\top \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \right) \\ & + \gamma^N \left(\mathbf{x}_N^\top P \mathbf{x}_N + \mathbf{p}^\top \mathbf{x}_N \right) && \text{(17a)} \\ \text{s.t.} \quad & \mathbf{x}_0 = \mathbf{s}, \qquad \mathbf{u}_0 = \mathbf{a}, && \text{(17b)} \\ & \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{b}, \qquad k \in \mathbb{I}_0^{N-1}, && \text{(17c)} \\ & C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \leq \mathbf{0}, \qquad k \in \mathbb{I}_0^{N-1}, && \text{(17d)} \\ & G\mathbf{x}_N + \mathbf{g} \leq \mathbf{0}, && \text{(17e)} \end{aligned}$$

where $\mathbf{z} := (\mathbf{x}_0, \mathbf{u}_0, \ldots, \mathbf{u}_{N-1}, \mathbf{x}_N)$. The dynamic constraints (17c) assume a nominal model without any perturbation. The tube-based approach then treats the system stochasticity, model uncertainties and safety constraint (16) by performing a suitable tightening of the path constraints (17d), i.e., $\mathbf{c}_k \geq \bar{\mathbf{c}}$ is used. The terminal constraints (17e) 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 [16]

$$V_{\boldsymbol{\theta}}(\mathbf{s}) := \min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}), \quad \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}) := \arg\min_{\mathbf{a}} Q_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}). \tag{18}$$

In practice $V_{\boldsymbol{\theta}}$ and $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ are computed jointly by solving (17) 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

$$H, \mathbf{h}, P, \mathbf{p}, A, B, \mathbf{b}, C, D, \bar{\mathbf{c}}, K, \boldsymbol{\theta}_{\mathbf{W}} \tag{19}$$

defining MPC scheme (17). Parameters K and $\boldsymbol{\theta}_{\mathbf{W}}$ are used to compute 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 $V_{\boldsymbol{\theta}}$, $Q_{\boldsymbol{\theta}}$, $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ 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 (17c), 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

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) = A\mathbf{s} + B\mathbf{a} + \mathbf{b} + \mathbf{W}_{\boldsymbol{\theta}} \supseteq \mathbf{S}_+(\mathbf{s}, \mathbf{a}). \tag{20}$$

Set $\mathbf{W}_{\boldsymbol{\theta}}$ is parametrized by parameter $\boldsymbol{\theta}_{\mathbf{W}}$. Common choices in robust MPC are to parametrize $\mathbf{W}_{\boldsymbol{\theta}}$ using ellipsoids or polytopes; in this paper we consider the latter.

In order to perform the required constraint tightening, i.e., the computation of $\mathbf{c}_k$, we introduce the prediction error of the nominal model (17c):

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

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

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

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

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

1) Path Constraints: We can now formalize the computation of the constraint tightening. Robust constraint satisfaction is obtained if

$$C\mathbf{x}_k + D\mathbf{u}_k + C_K\mathbf{E}_k + \bar{\mathbf{c}} \leq \mathbf{0}, \tag{21}$$

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

$$\begin{aligned} \mathbf{d}_{i,k} &:= \max_{\mathbf{e}} \ (C_K)_i \mathbf{e} && \text{s.t. } \mathbf{e} \in \mathbf{E}_k \\ &= \max_{\mathbf{w}} \ (C_K)_i \sum_{j=0}^{k-1} (A_K)^j \mathbf{w}_j && \text{s.t. } \mathbf{w}_j \in \mathbf{W}_{\boldsymbol{\theta}}. \end{aligned} \tag{22}$$

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 (22) can be formulated as an LP; this implies that: (a) constraint tightening is relatively cheap to compute; (b) as detailed in Section IV, the SDC enforcement becomes easier to derive.

2) Terminal Constraint: In order to guarantee that Problem (17) 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 $\mathcal{X}_{\mathrm{f}} := \{\, \mathbf{x} \,|\, G\mathbf{x} + \mathbf{g} \leq 0 \,\}$ must be such that there exists a terminal control law κ_{f} which guarantees that the state error $\mathbf{e}$ does not diverge and all path constraints are always satisfied. While in principle it is possible to choose any law κ_{f}, here we consider $\kappa_{\mathrm{f}} = -K\mathbf{e}_k$.

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

$$\begin{aligned} \mathcal{X}_0 &:= \{\ \mathbf{x}_0 \mid C\mathbf{x}_0 + D\mathbf{u}_0 + \mathbf{c}_0 \leq 0 \ \}, \\ \mathcal{X}_k &:= \{\ \mathbf{x}_k \mid C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \leq 0, \text{ (17c) } \}. \end{aligned}$$

Note that, by (21)-(22), $\mathbf{x}_k \in \mathcal{X}_k$ implies

$$C\mathbf{s}_k + D(\mathbf{u}_k - K\mathbf{e}_k) + \bar{\mathbf{c}} \leq 0.$$

Set $\mathcal{X}_\infty$ is Robust Positive Invariant (RPI) [23], i.e.,

$$\mathbf{x}_k \in \mathcal{X}_\infty \quad \Rightarrow \quad \mathbf{x}_{k+1} \in \mathcal{X}_\infty, \quad C\mathbf{s}_j + D\mathbf{u}_j + \bar{\mathbf{c}} \leq 0, \, \forall\, j > 0.$$

Additionally, whenever the system is stable and the origin is in the interior of the constraint set, the RPI is finitely determined [23, Theorem 6.3], i.e., $\exists\, k' < \infty$ s.t. $\mathcal{X}_{k'} \equiv \mathcal{X}_{k'+i}$, $i = 1, 2, \dots$. The stability requirement further motivates the introduction of feedback through matrix K.

Consistently with the previously used notation, we define

$$G := \begin{bmatrix} C_K \\ C_K A_K \\ \vdots \\ C_K A_K^{k'} \end{bmatrix}, \qquad \bar{\mathbf{g}} := \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{k'} \end{bmatrix}, \tag{23}$$

where we stress that $\mathbf{c}_k = \bar{\mathbf{c}} + \mathbf{d}_k$, such that, by using $\kappa_{\mathrm{f}} = -K\mathbf{e}$ one can spare a large amount of computations, $\bar{\mathbf{g}}$ being already computed. The condition $G\mathbf{x}_N + \bar{\mathbf{g}} \leq 0$ 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 must be tightened. Analogously to the case of path constraints, we define $\mathbf{g} = \bar{\mathbf{g}} + \mathbf{h}$ with

$$\mathbf{h}_{i,k} := \max_{\mathbf{w}} \ G_i \sum_{j=0}^{k'-1} A_K^j \mathbf{w}_j \qquad \text{s.t. } \mathbf{w}_j \in \mathbf{W}_{\boldsymbol{\theta}}. \tag{24}$$

Remark 2: Typically, constraints that can never become active are removed from (23), so as to reduce the dimension.

3) Safety Property: Safety is guaranteed by the following result on tube MPC.

Proposition 1 (Recursive Feasibility): Assume that $\mathcal{X}_{\mathrm{f}} := \{\, \mathbf{x} \,|\, G\mathbf{x} + \mathbf{g} \leq 0 \,\}$ is RPI and Problem (17) is feasible at time $k = 0$. Then Problem (17) is feasible for all $\mathbf{w}_k \in \mathbf{W}_{\boldsymbol{\theta}}$ and all times $k \geq 0$. If moreover (20) holds, the real system (1) satisfies the safety constraint (16) at all times $k \geq 0$.

Proof: The proof can be found in, e.g., [20]. ∎

While the framework of robust linear MPC based on RPI 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, K, but also all other parameters of the MPC formulation (19). In order to compute $\nabla_{\boldsymbol{\theta}}\mathbf{c}_k$, $\nabla_{\boldsymbol{\theta}}C_N$, $\nabla_{\boldsymbol{\theta}}\mathbf{c}_N$ one can use results from parametric optimization to obtain the following lemmas [24].

Consider a parametric NLP with cost $\phi_{\boldsymbol{\theta}}$, primal-dual variable $\mathbf{y}$ and parameter $\boldsymbol{\theta}$. We refer to [25] 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., [24]. ∎

Corollary 1: Assume that LICQ, SOSC and SC hold at the optimal solution of (17). 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:

$$\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}(\mathbf{s}) = \nabla_{\boldsymbol{\theta}} \bar{l}_{\boldsymbol{\theta}}(\mathbf{y}), \quad \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) = \nabla_{\boldsymbol{\theta}} l_{\boldsymbol{\theta}}(\mathbf{y}), \tag{25a}$$

$$\frac{\partial \mathbf{r}_{\boldsymbol{\theta}}}{\partial \mathbf{y}}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\mathbf{y}^\star = \frac{\partial \mathbf{r}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}, \tag{25b}$$

where $l_{\boldsymbol{\theta}}$ is the Lagrangian of Problem (17), $\bar{l}_{\boldsymbol{\theta}}$ is the Lagrangian when constraint $\mathbf{u}_0 = \mathbf{a}$ is eliminated, and $\mathbf{r}_{\boldsymbol{\theta}}$ denotes the 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 (25a).

In (25), $\mathbf{r}_{\boldsymbol{\theta}}$, $l_{\boldsymbol{\theta}}$, and $\bar{l}_{\boldsymbol{\theta}}$ depend on $\mathbf{c}_k$, $\mathbf{g}$ which, in turn, depend on $\boldsymbol{\theta}$ as they are optimal values of parametric optimization problems (22) and (24). Consequently, one needs to evaluate $\nabla_{\boldsymbol{\theta}}\mathbf{c}_k$, $\nabla_{\boldsymbol{\theta}}\mathbf{g}$. In the following, we further detail the application of Lemma 1 to this case.

We consider only Problem (22), since the derivation for (24) 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

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

$$\mathbf{d}^j_{i,k} := \max_{\mathbf{w}_j} \ (C_K)_i A^j_K \mathbf{w}_j \qquad \text{s.t.} \qquad \mathbf{w}_j \in \mathcal{W}_{\boldsymbol{\theta}}. \tag{26}$$

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

Then, Lemma 1 can be applied to obtain

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

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

where $l^{\mathbf{d}}_{\boldsymbol{\theta},i,j}$ is the Lagrangian of Problem (26). Provided that a second-order method is used for solving Problem (26), then the matrix factorization is available and can be reused to compute the sensitivities at a negligible cost. The chain rule yields

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

Remark 4: As underlined above, Problems (26) 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 (26) are very low dimensional and are therefore solved extremely quickly.

C. Guaranteeing Feasibility and LICQ

We illustrate next two issues that can be easily encountered when deploying RL based on MPC (17). Since these are commonly encountered, a simple solution which has become a standard in MPC is readily available.

a) 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 (37)), but recursive feasibility is potentially lost, unless some form of relaxation is deployed for the path constraints.

b) Sensitivity Computation: The sensitivity computation is valid if LICQ holds. However, Problem (17) is not guaranteed to satisfy LICQ, as illustrated by the following example.

We propose to address both issues by using a common trick in MPC, i.e., a constraint relaxation for Constraints (17d) and (17e) with an exact penalty [26]: variables $\boldsymbol{\sigma}_k$ are introduced, the constraints are modified as

$$C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \le \boldsymbol{\sigma}_k, \tag{27a}$$

$$G\mathbf{x}_N + \mathbf{g} \le \boldsymbol{\sigma}_N, \qquad \boldsymbol{\sigma}_k \ge 0, \qquad k \in \mathbb{I}_1^N, \tag{27b}$$

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 can not be guaranteed even if the relaxation proposed above is deployed.

We formalize the result in the next proposition.

Proposition 2: Assume that Constraints (17d) and (17e) are relaxed as per (27) 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 [26] and [27, Theorem 14.3.1]. Regarding LICQ, we first note that in a formulation without Constraints (17d) and (17e) LICQ holds by construction: Constraints (17b) and (17c) can be eliminated by condensing [28], 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 (17d) or (17e) with Jacobian $\boldsymbol{\nu}$. By introducing slack variable σ, the new Jacobian becomes $\begin{bmatrix} \boldsymbol{\nu} & 1 \end{bmatrix}$, which is by construction linearly independent with $\begin{bmatrix} \boldsymbol{\nu} & 0 \end{bmatrix}$ and, consequently, with the other constraints in the problem. ■

IV. Safe Design Constraint and Big Data

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 is collected at high rates. In order to be able to deal with such amounts of data in real-time it is necessary to (a) retain only strictly relevant data, and (b) 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:

$$\mathcal{D} = \{(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2), \ldots, (\mathbf{s}_n, \mathbf{a}_n, \mathbf{s}_{n+1})\}. \tag{28}$$

The problem of enforcing the SDC (12) is related to that of estimating the dispersion set $\mathbf{S}_+$ which has been studied in the context of *set-membership system identification* (SMSI) [29]. Essentially, the dispersion set must satisfy

$$\mathbf{s}_+ \in \hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}), \qquad \forall\ (\mathbf{s}, \mathbf{a}, \mathbf{s}_+) \in \mathcal{D}, \tag{29}$$

i.e., $\boldsymbol{\theta}$ must satisfy the SDC (12), equivalently stated as:

$$\mathcal{S}_{\mathcal{D}} := \{\boldsymbol{\theta} \,|\, \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{u}_k) \leq 0,\, \forall\, (\mathbf{s}_+, \mathbf{s}, \mathbf{u}) \in \mathcal{D}\}. \tag{30}$$

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 $\mathbf{s}, \mathbf{a}$, parametrized 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 parameter $\boldsymbol{\theta}$ is 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 optimal for the control task iff $\nexists\ \bar{\boldsymbol{\theta}}$ s.t. $J(\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}) < J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$.

In other words, the optimal parameter minimizes the cost subjecto to the SDC (12). Based on this definition, the optimal set approximation is obtained by letting RL adapt $\boldsymbol{\theta}$ to maximize performance. Every time the RL problem (15) 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.

Consistently with Section III-A, we choose $\mathbf{g}_{\boldsymbol{\theta}}$ affine, i.e.,

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

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 $M, \mathbf{m}$ for the two sets coincide. Therefore, the SDC directly defines the uncertainty set $\mathbf{W}_{\boldsymbol{\theta}}$ used by MPC to compute safe policies.

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 (17c) 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 parametrization and MPC formulation, an *optimal data compression* selects a dataset $\bar{\mathcal{D}} \subseteq \mathcal{D}$ such that:

$$\left|\bar{\mathcal{D}}\right| = \min_{\hat{\mathcal{D}}} \ \left|\hat{\mathcal{D}}\right| \qquad \text{s.t.}\ \ \mathcal{S}_{\hat{\mathcal{D}}} \equiv \mathcal{S}_{\mathcal{D}}. \tag{31}$$

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

The introduction of the nominal model (17c) allows us to restructure the data and only store the noise $\mathcal{W} := \{\mathbf{w}_0, \ldots, \mathbf{w}_n\}$, obtained from

$$\mathbf{w}_k = \mathbf{s}_{k+1} - (A\mathbf{s}_k + B\mathbf{a}_k + \mathbf{b}). \tag{32}$$

By using (32), we rewrite the SDC as

$$\mathcal{S}_{\mathcal{D}} = \mathcal{S}_{\mathcal{W}} := \{\boldsymbol{\theta} \,|\, M\mathbf{w} \leq \mathbf{m},\, \forall\, \mathbf{w} \in \mathcal{W}\}, \tag{33}$$

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 $n_{\mathbf{s}}^2 n_{\mathbf{a}}$ 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 (33) is then

$$\hat{\mathbf{S}}_+(\mathbf{s}, \mathbf{a}, \boldsymbol{\theta}) = \{\, A\mathbf{s} + B\mathbf{a} + \mathbf{b} + \mathbf{w} \,|\, \forall\, \mathbf{w} \,\text{s.t.}\, M\mathbf{w} \leq \mathbf{m} \,\}, \tag{34}$$

such that (22), (24) used in constraint tightening are LPs. Additionally, it is cheap to (a) evaluate if $\mathbf{s}_+ \in \hat{\mathbf{S}}_+$ by using (32) and (b) 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 (a) requires a single evaluation of the inequality in (34), (b) requires to evaluate the inequality for each data point in (33).

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 (33), 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

$$\zeta := \min_{\mathbf{z}} \quad \sum_{i=1}^{n} \mathbf{z}_i \quad \text{s.t.} \quad \hat{\mathbf{w}} = \sum_{i=1}^{n} \mathbf{z}_i \mathbf{w}_i, \quad \mathbf{z} \geq 0, \tag{35}$$

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 (35).

Proof: The definition of convex hull implies $\hat{\mathbf{w}} \in \mathrm{Conv}(\mathcal{W}) \Rightarrow \exists\ \mathbf{z} \geq 0, \|\mathbf{z}\|_1 = 1$ s.t. $\hat{\mathbf{w}} = \sum_{i=1}^{n} z_i \mathbf{w}_i$, 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 $\zeta \mathbf{w}_i \in \mathrm{Conv}(\mathcal{W}),\ \forall\ \zeta \in [0, 1]$. The case $\zeta < 1$ is thus immediately reconducted to the case $\zeta = 1$. ∎

Note that (35) needs to be solved every time a new sample is available in order to keep the convex hull updated. Formulation (35) 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. 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 $A, B, \mathbf{b}$.

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 (34), 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 (31). ∎

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 (12). 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 conservativeness in MPC (17) 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 big data. Moreover, the parametrization of $\mathbf{W}_{\boldsymbol{\theta}}$ should be selected consistently with the algorithm used for solving the robust MPC Problem. Convex polytopes can be parametrized using the so-called (a) facet or (b) vertex representation. In case (a), the set is parametrized as $\mathbf{W}_{\boldsymbol{\theta}} := \{\, \mathbf{w} \,|\, M\mathbf{w} \leq \mathbf{m} \,\}$ with parameter $\boldsymbol{\theta}_{\mathbf{W}} = (M, \mathbf{m})$. In case (b), the set is parametrized 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 (33), which becomes

$$\mathcal{S}_{\bar{\mathcal{W}}} = \left\{ \boldsymbol{\theta} \,|\, M\mathbf{w} \leq \mathbf{m},\, \forall\, \mathbf{w} \in \bar{\mathcal{W}} \right\}. \tag{36}$$

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 6: In any sample-based context it is possible that a new sample 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 (12). A straightforward adaptation of $\mathcal{S}_{\bar{\mathcal{W}}}$ is obtained as

$$\mathbf{m}_i \leftarrow \max(\mathbf{m}_i, M_i \mathbf{w}_k). \tag{37}$$

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 (17) such that it is necessary to deploy the constraint relaxation proposed in Section III-C.

Remark 7: As data are collected, $|\mathcal{D}|$ increases 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. Such investigations are beyond the scope of this paper and require future research.

Remark 8: The proposed analysis assumes for simplicity that the model is fixed, since one must take extra care if the model parameters A, B are updated. Indeed, if one updates $A, B, \mathbf{b}$ to $\tilde{A}, \tilde{B}, \tilde{\mathbf{b}}$, 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 pair. Moreover, when using the convex hull approach, one must additionally recompute the convex hull. 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 $\tilde{\mathcal{W}} = \mathcal{W} + \mathbf{b} - \tilde{\mathbf{b}}$. Finally, any update in $A, B, \mathbf{b}$ has an impact on $V_\theta, Q_\theta, \boldsymbol{\pi}_\theta$, such that additional care is required.

V. Safe RL

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

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha(\boldsymbol{\theta}^* - \boldsymbol{\theta}_k), \tag{38}$$

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 RL problem is constrained, see (15). Moreover, for MPC to be properly formulated we require its cost function to be positive-definite. We then consider a framework similar to the one proposed in [17], with $\boldsymbol{\theta}^*$ solution of

$$\min_{\boldsymbol{\theta}} \ \psi(\mathbf{s}_{k+1}, \mathbf{s}_k, \mathbf{a}_k, \boldsymbol{\theta}), \tag{39a}$$

$$\text{s.t. } H \succ 0, \qquad P \succ 0, \tag{39b}$$

$$M\mathbf{w} \leq \mathbf{m}, \qquad \forall\, \mathbf{w} \in \bar{\mathcal{W}}. \tag{39c}$$

The SDC (12) is imposed in (39c) and, similarly to [17], we introduce (39b) to make sure that the MPC cost is convex and, therefore, easily and efficiently solvable.

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: (a) updating $\boldsymbol{\theta}$ only when feasible, (b) reducing the step size until feasibility is recovered, (c) enforcing feasibility as an additional constraint in (39). It is worth mentioning that we did not encounter this issue in the simulations we performed. Nevertheless, future research will investigate this issue further.

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 non-smooth. This is the case when SC does not hold either in the MPC problem (17) or in the constraint tightening problems (22), (24), 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 paper 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 (17). 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 (18). 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

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

This issue can be tackled by using a modified version of the robust MPC Problem (17):

$$\min_{\mathbf{x},\mathbf{u}} \ \ (17\text{a}) + f\,(\mathbf{u}_0, \mathbf{q}) \tag{40a}$$

$$\text{s.t. } \ \mathbf{x}_0 = \mathbf{s}, (17\text{c}), (17\text{d}), (17\text{e}), \tag{40b}$$

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 $\mathbf{q}$. While some issues could arise when doing constrained exploration, this topic is beyond the scope of this paper and discussed in [18], [19].

We can now state the main result of this paper.

Theorem 2: Consider an RL scheme where (a) the robust MPC (17) is used as function approximator; (b) the available data is handled as detailed in Section IV; and (c) exploration is performed according to (40). Assume that all new data

yields $\mathbf{w}_k \in \bar{\mathcal{W}}$. Then, the RL scheme is safe in the sense of Definition 1, and optimal in the sense of Definitions 2 and 3.

Proof: Since $\mathbf{w}_k \in \bar{\mathcal{W}}$, by Proposition 1 robust MPC (17) is recursively feasible. Moreover, exploration is performed using (40), i.e., (17) with a modified cost, such that recursive feasibility is preserved. Set membership optimality (Definition 2) is obtained by construction, since this is the objective of RL. Data compression optimality (Definition 3) is a direct consequence of Theorem 1. ■

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 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 quickly recovered, as the RL parameter is instantaneously adjusted to account for the new sample. As also highlighted in Remark 6 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 paper and will be the subject of future research.

2) Approximation Quality: It has been proven in [16, Theorem 1] that, provided that the MPC parametrization is rich enough, the correct value and action-value functions can be recovered exactly. In the context of this paper, however, the parametrization 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 parametrization of [16], while $\boldsymbol{\theta}_{\mathbf{W}}$ is the parameter vector parametrizing function $\mathbf{g}_{\boldsymbol{\theta}} := \mathbf{g}_{\boldsymbol{\theta}_{\mathbf{W}}}$. Then, provided that the parametrization 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 (17). Therefore, as opposed to the result of [16], the equivalence only holds on the subset of the feasible domain of the RL problem which can be described by the chosen parametrization of $\mathbf{g}_{\boldsymbol{\theta}}$.

Since the hypothesis of a perfect parametrization 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 parametrization, 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 manipluating $\boldsymbol{\theta}$. 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. This topic is investigated in [18], [19].

3) Model Adaptation: In principle, one could choose to let RL update any of the parameters (19) of the MPC scheme (17), including the model parameters A, B, $\mathbf{b}$. The model used by MPC plays an important role in (a) the definition of the cost function and (b) the quality of the uncertainty set approximation $\mathbf{W}_{\boldsymbol{\theta}}$. As observed in Remark 8, 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.

In [16, Theorem 1] and [17, Corollary 2] it is proven that RL can find $Q_{\boldsymbol{\theta}} = Q$, $V_{\boldsymbol{\theta}} = V$ and $\boldsymbol{\pi}_{\boldsymbol{\theta}} = \boldsymbol{\pi}$, provided that the parametrization is descriptive enough. However, typically the parametrization 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 parametrization 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: stability to the minimum RPI (mRPI) is proven, under the assumptions of having $\gamma = 1$ and a positive-definite stage cost, usually called *tracking* cost, as opposed to *economic* cost. While the literature on RL does not discuss the properties of the cost function (or rather reward function in RL), the distinction is relevant in relation to the stability properties of the closed-loop system and a thorough discussion on economic MPC can be found in [30], [31], [32]. The linear-quadratic case, which is particularly relevant for our framework, has been analyzed in [33], [34], [35], [36], [37]. We only recall here that all the conclusions drawn in [16] apply directly to our setup, i.e., the economic case can be reconducted to the tracking case by additionally learning a storage function used to rotate the cost. The main difficulty we are left to deal with is the presence of a discount factor in the MPC formulation, which complicates the stability analysis. While one could foresee formulating (17) 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. By exploiting the results of [33] it should be straightforward to prove that stabilizing linear control laws can be recovered exactly. However, a thorough investigation of this topic is the subject of ongoing research.

VI. Simulation Results

We demonstrate the theoretical developments with the following simple example in dimension 2, 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 $-\mathbf{1} \leq \mathbf{s} \leq \mathbf{1}$, $-10 \leq a \leq 10$. 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 parametrize $\mathbf{W}_{\boldsymbol{\theta}}$ as a polytope with 4 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 paper, 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 [36], [35], [38] in the context of economic MPC, concluding that under suitable assumptions verified by the system considered here, practical stability is obtained.

We update $\boldsymbol{\theta}$ according to (38)-(39) with $\alpha = 0.1$, ψ given by (6) (Q learning) and M, $\mathbf{m}$ elements in $\boldsymbol{\theta}$. Problem (39) is solved to full convergence at each step such that: (a) the use of globalization techniques guarantees an improvement in the sense that $\boldsymbol{\theta}^*$ is a better fit than $\boldsymbol{\theta}_k$ for the sample at hand; (b) positive-definiteness of the cost yields a well-posed MPC formulation; and (c) 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 Figures 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 towards the reference and 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 Figure 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 4 facets.

We ran an additional simulation in which we let RL also adapt 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 author's 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 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.

VII. Conclusions and Future Work

In this paper 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 which makes it possible to deal with very large amounts of data in real-time.

The proposed framework paves the road for several extensions: (a) one can easily foresee the use of robust MPC based on scenario-trees (which is also suitable for nonlinear models); (b) the use of stochastic or deterministic policy gradient is expected to further improve performance, as discussed in Section II; (c) 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 and Notes

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

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "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," *Trans. Sys. Man Cyber Part C*, 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 *In Advances in Neural Information Processing Systems 19*. MIT Press, 2007, p. 2007.

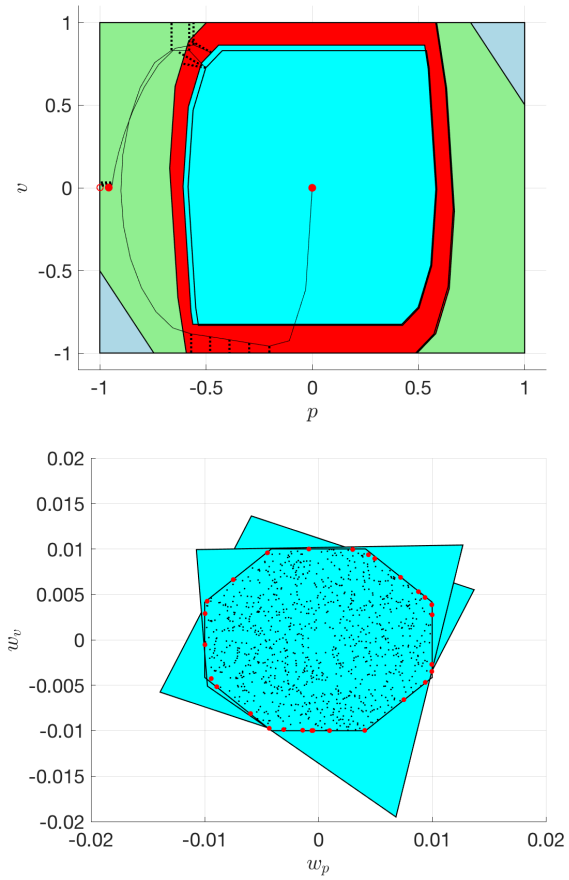

Fig. 2: Snapshots at $k = 0$ and $k = 24$. Top figure (state space): state constraint set (light blue), state constraint with feedback K (green), RPI set (red), terminal constraint set $\mathcal{X}_\mathrm{f}$ (cyan for $k = 24$, transparent for $k = 0$), initial state (red dot), predicted trajectory (solid black line), reference $\mathbf{s}^\mathrm{r}$ (red circle), tightening of the active constraints (dashed black line). Bottom figure (noise space): true uncertainty set (transparent octagon), noise samples (black dots), vertices of their convex hull (red dots), 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 $\mathcal{X}_\mathrm{f}$.

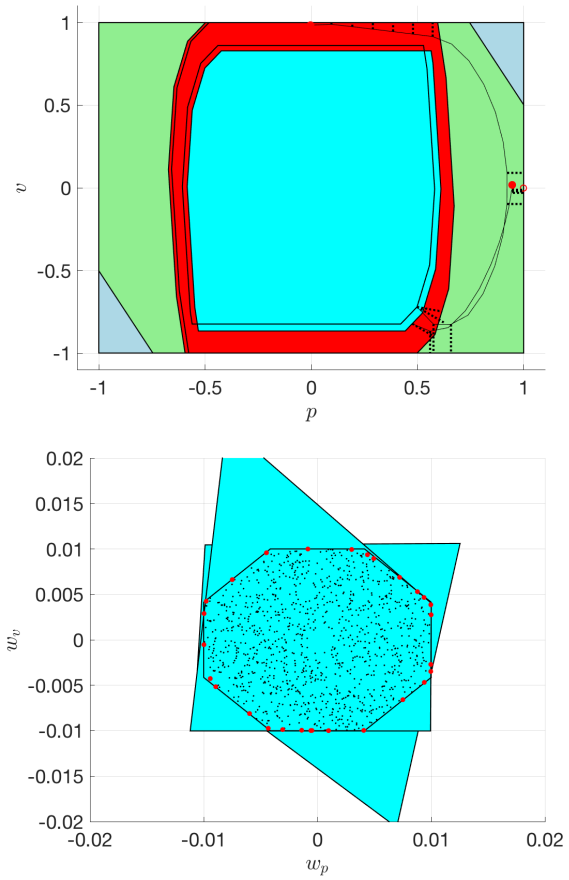

Fig. 3: Snapshots at $k = 34$ and $k = 109$: same convention as Figure 2. Both the RPI and terminal sets moved closer to the setpoint by a better approximation $\mathbf{W}_{\boldsymbol{\theta}}$ for the specific control task (see the bottom plot). Moreover, next to $\mathbf{s}^\mathrm{r}$ the constraints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

[5] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ser. NIPS'99. Cambridge, MA, USA: MIT Press, 1999, pp. 1057–1063.

[6] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, ser. ICML'14, 2014, pp. I–387–I–395.

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

[8] J. F. J. Garcia, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, pp. 1437–1480, 2013.

[9] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based Model Predictive Control for Safe Exploration and Reinforcement Learning," 2018, published on Arxiv.

[10] 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.

[11] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.

[12] 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. Curran Associates, Inc., 2017, pp. 908–918.

[13] R. Murray and M. Palladino, "A model for system uncertainty in reinforcement learning," *Systems & Control Letters*, vol. 122, pp. 24 – 31, 2018.

[14] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[15] 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 Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[16] S. Gros and M. Zanon, "Data-Driven Economic NMPC using Reinforcement Learning," *IEEE Transactions on Automatic Control*, 2018, (in press).

[17] M. Zanon, S. Gros, and A. Bemporad, "Practical Reinforcement Learning of Stabilizing Economic MPC," in *Proceedings of the European Control Conference*, 2019, (accepted).

[18] S. Gros and M. Zanon, "Towards Safe Reinforcement Learning Using NMPC and Policy Gradients - Stochastic case (Part I)," *IEEE Transactions on Automatic Control*, 2019, (submitted).

[19] ——, "Towards Safe Reinforcement Learning Using NMPC and Policy Gradients - Deterministic case (Part II)," *IEEE Transactions on Automatic Control*, 2019, (submitted).

[20] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 2001.
[21] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.
[22] D. Mayne, "Robust and stochastic mpc: Are we going in the right direction?" *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 1 – 8, 2015, 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015.
[23] 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.
[24] C. Büskens and H. Maurer, *Online Optimization of Large Scale Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, ch. Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems, pp. 3–16.
[25] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.
[26] P. Scokaert and J. Rawlings, "Feasibility Issues in Linear Model Predictive Control," *AIChE Journal*, vol. 45, no. 8, pp. 1649–1659, 1999.
[27] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester: Wiley, 1987.
[28] 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: Birkhäuser, 1983, pp. 95–121.
[29] D. Bertsekas and I. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," vol. 16, pp. 117–128, 1971.
[30] M. Diehl, R. Amrit, and J. Rawlings, "A Lyapunov Function for Economic Optimizing Model Predictive Control," *IEEE Trans. of Automatic Control*, vol. 56, no. 3, pp. 703–707, March 2011.
[31] R. Amrit, J. Rawlings, and D. Angeli, "Economic optimization using model predictive control with a terminal cost," *Annual Reviews in Control*, vol. 35, pp. 178–186, 2011.
[32] M. A. Müller, D. Angeli, and F. Allgöwer, "On necessity and robustness of dissipativity in economic model predictive control," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1671–1676, 2015.
[33] M. Zanon, S. Gros, and M. Diehl, "Indefinite Linear MPC and Approximated Economic MPC for Nonlinear Systems," *Journal of Process Control*, vol. 24, pp. 1273–1281, 2014.
[34] ——, "A Tracking MPC Formulation that is Locally Equivalent to Economic MPC," *Journal of Process Control*, 2016.
[35] M. Zanon and T. Faulwasser, "Economic {MPC} without terminal constraints: Gradient-correcting end penalties enforce asymptotic stability," *Journal of Process Control*, vol. 63, pp. 1 – 14, 2018.
[36] T. Faulwasser and M. Zanon, "Asymptotic Stability of Economic NMPC: The Importance of Adjoints," in *Proceedings of the IFAC Nonlinear Model Predictive Control Conference*, 2018.
[37] L. Grüne and R. Guglielmi, "Turnpike properties and strict dissipativity for discrete time linear quadratic optimal control problems," *SIAM Journal on Control and Optimization*, vol. 56, no. 2, pp. 1282–1302, 2018.
[38] L. Grüne, "Economic receding horizon control without terminal constraints," *Automatica*, vol. 49, pp. 725–734, 2013.

Sébastien Gros received his Ph.D degree from EPFL, 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 university of KU Leuven, where his main research focus was on optimal control and fast NMPC for complex mechanical systems. He joined the Department of Signals and Systems at Chalmers University of Technology, Göteborg in 2013, where he became associate Prof. in 2017. He is now full Prof. at NTNU, Norway and affiliate Prof. at Chalmers. His main research interests includes numerical methods, real-time optimal control, reinforcement learning, and the optimal control of energy-related applications.

Mario Zanon received the Master's degree in Mechatronics from the University of Trento, and the Diplôme d'Ingénieur from the Ecole Centrale Paris, in 2010. After research stays at the KU Leuven, University of Bayreuth, Chalmers University, and the University of Freiburg he received the Ph.D. degree in Electrical Engineering from the KU Leuven in November 2015. He held a Post-Doc researcher position at Chalmers University until the end of 2017 and is now Assistant Professor at the IMT School for Advanced Studies Lucca. His research interests include numerical methods for optimization, economic MPC, reinforcement learning, and the optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.