Learning for MPC with Stability & Safety Guarantees ⋆

Sebastien Gros [a], Mario Zanon [b]

[a] *Dept. of Cybernetics, Faculty of Information Technology, NTNU, Norway*

[b] *IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy*

Abstract

The combination of learning methods with Model Predictive Control (MPC) has attracted a significant amount of attention in the recent literature. The hope of this combination is to reduce the reliance of MPC schemes on accurate models, and to tap into the fast developing machine learning and reinforcement learning tools to exploit the growing amount of data available for many systems. In particular, the combination of reinforcement learning and MPC has been proposed as a viable and theoretically justified approach to introduce explainable, safe and stable policies in reinforcement learning. However, a formal theory detailing how the safety and stability of an MPC-based policy can be maintained through the parameter updates delivered by the learning tools is still lacking. This paper addresses this gap. The theory is developed for the generic Robust MPC case, and applied in simulation in the robust tube-based linear MPC case, where the theory is fairly easy to deploy in practice. The paper focuses on Reinforcement Learning as a learning tool, but it applies to any learning method that updates the MPC parameters online.

Key words: Safe MPC Learning, Safe MPC-based policies, Safe Reinforcement Learning, Robust MPC, Stability

1 Introduction

Model Predictive Control (MPC) is a very successful tool for generating policies that minimize a certain cost under some state and input constraints. MPC uses model-based predictions of the future system trajectories to produce a control input profile over a future time window that satisfies the constraints while minimizing the cost. Closed-loop control policies are then obtained by updating that control input profile at every time instant, in a receding-horizon fashion, based on the latest state of the system and information on its environment. MPC heavily relies on a model of the system at hand to perform well. However, accurate models are expensive to develop, and can be too complex to use in the context of MPC. As a result, while MPC can deliver a reasonable approximation of the optimal policy, it is usually suboptimal.

In the recent literature, various learning techniques, often borrowed from the field of machine learning, have been investigated to address this problem. A number of them focus on using machine learning to improve the fitting of the MPC model to the data, see, e.g., [11] and references therein. Other approaches argue for adjusting the MPC model, cost and constraints in view of maximizing directly the closed-loop performance. The authors of [9] provide strong theoretical justifications for this approach, and propose to use Reinforcement Learning (RL) techniques to perform that adjustment in practice. The use of learning techniques within control has been proposed in, e.g., [3,4,14–16,21,23,2,9,31,33,18,34,1,30,27].

A prime motivation for using MPC-based policies is the possibility to enforce constraints on the system trajectories. This feature can be leveraged to ensure the safety of the system at hand by defining constraints that limit its evolution to a safe set, and including these constraints in the MPC formulation. When using inaccurate models, or when the real system is stochastic, safety can be maintained via robust MPC techniques, where the uncertainties are taken into account in the MPC formulation. Robust MPC delivers policies that are safe by construction, using a worst-case approach, and ensures that the real system trajectories satisfy the constraints at all

⋆ This paper was not presented at any IFAC meeting. Corresponding author M. Zanon. This paper was partially supported by the Italian Ministry of University and Research under the PRIN'17 project "Data-driven learning of constrained control systems" , contract no. 2017J89ARP; and by the Norwegian Research Council project "Safe Reinforcement-Learning using MPC" (SARLEM).

Email addresses: sebastien.gros@ntnu.no (Sebastien Gros), mario.zanon@imtlucca.it (Mario Zanon).

time.

The combination of learning with robust MPC has been investigated in [31]. This combination arguably offers the most direct pathway to optimize the closed-loop performance of an MPC-based policy while maintaining its safety. The authors of [31] additionally argue that robust MPC offers a direct pathway towards safe reinforcement learning, providing strong certificates of safety. Indeed, when using robust MPC, the policies produced via learning can be made safe and stable by construction by imposing certain constraints on the parameter updates suggested by the learning algorithm [31]. In contrast, in classic reinforcement learning, e.g., based on Deep Neural Networks (DNN), enforcing the safety of the resulting policy is typically done via extensive in silico validations using Monte Carlo techniques. In that context, formal certificates of strong safety require in principle an infinite amount of simulations and are therefore difficult to establish in practice.

While [31] details how to learn policies that are safe and stable by construction, an important gap remains to be addressed. Performing learning on a control policy requires that regular updates of the policy parameters are implemented. Some learning methods implement the updates at every sampling time of the policy, while other methods implement the updates less frequently. If the parameter updates are to be implemented while the system is being operated, then implementing safe and stable policies at every parameter update does not necessarily yield an overall safe and stable closed-loop system. Indeed, safety with parameter updates can only be guaranteed if the update takes place when the system state is within specific sets. Stability is also not guaranteed when the parameters are frequently updated, even if each policy implemented is stable. Ensuring safety and stability through the parameter updates then requires additional conditions. This paper addresses that issue by detailing how to maintain safety and stability through the parameter updates, and therefore provides a complete theoretical framework to deploy safe and stable learning for MPC. The paper focuses on learning techniques based on reinforcement learning, but applies equally to all learning techniques that propose directions in the MPC parameters space along which the MPC parameters ought to be updated.

The paper is organized as follows. Section 2 provides background material on MDPs. Section 3 proposes a definition of safe policies and Section 4 provides background material on robust MPC. Section 5 presents conditions for building a safe combination of RL and robust MPC, where the parameter updates provided by RL do not jeopardize the safety of the robust MPC scheme. Section 6 presents conditions for RL to update the MPC parameters while maintaining the system stability, discussed in an augmented parameter-state space. Section 7 illustrates the theoretical results by means of two simple examples and Section 8 concludes the paper.

2 Background

We consider real systems that can be described as discrete-time Markov chains with continuous state and action spaces. We will label the underlying conditional transition probability density over the states $\mathbf{s}$ and actions $\mathbf{a}$ as:

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

where n and m are the state and input space sizes, respectively. Throughout the paper, index i will refer to the physical time of the system. We will assume that (1) is only inaccurately known. We will assume in the following that a stage cost

$$L\left(\mathbf{s}_i,\mathbf{a}_i\right)\,:\,\mathbb{R}^n\times\mathbb{R}^m\,\rightarrow\mathbb{R} \tag{2}$$

is provided, and that our goal is to find the parameters $\boldsymbol{\theta}$ of a policy

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}\,:\,\mathbb{R}^n\rightarrow\mathbb{R}^m \tag{3}$$

delivering the actions, i.e., $\mathbf{a}_i=\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}_i\right)$, so as to minimize the expected discounted cost:

$$J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)=\mathbb{E}\left[\left.\sum_{i=0}^{\infty}\gamma^iL\left(\mathbf{s}_i,\mathbf{a}_i\right)\,\right|\,\mathbf{a}_i=\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}_i\right)\right], \tag{4}$$

where $\mathbb{E}[\cdot]$ is the expected value operator applying to the real trajectories yielded by (1) in closed-loop with policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and $\gamma\in(0,1]$ a discount factor. We will consider that the actions delivered by the policy are possibly restricted to a subset of $\mathbb{R}^m$, i.e., the minimization of J is subject to

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}_i\right)\in\mathbb{P},\quad\forall\,\mathbf{s}_i. \tag{5}$$

Finding the parameters $\boldsymbol{\theta}$ that (locally) minimize the closed-loop cost $J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$, and therefore maximize the closed-loop performance of the policy is arguably one of the main goals of any learning algorithms focusing on improving a policy. E.g., many methods in Reinforcement Learning (RL) deal with the evaluation of the policy gradient $\nabla_{\boldsymbol{\theta}}J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$, which is used to update the policy parameters $\boldsymbol{\theta}$ such that $J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$ is sequentially decreased. In this paper, we will focus for simplicity on parameter updates that reduce $J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$ directly, albeit a number of learning techniques compute updates that are not directly based on $J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$. The theory presented here applies to all learning methods for MPC with minor modifications.

We consider in this paper that we seek policies that keep the system safe in the sense of respecting some state constraints expressed as:

$$\mathbf{s}_i \in \mathcal{X}, \tag{6}$$

for any time $i = 0, \ldots, \infty$, where $\mathbf{s}_{0,\ldots,\infty}$ denotes the closed-loop trajectories with policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. Note that for the sake of simplicity, we will not treat mixed state-input constraints here, even though the proposed results arguably readily extend to that case. Throughout the paper, we will assume that the real state transition (1) is imperfectly or only coarsely known, and difficult to capture via simple mathematical models.

3 Safe Policies

Ideally, a policy is safe if (6) holds at all time i with probability one. Unfortunately, guaranteeing this unitary probability is impossible without a perfect knowledge of the system dynamics (1). In that context, a more realistic notion of safety can be defined in the context of Bayesian inference, where safety is regarded as probabilistic, conditioned on our knowledge of the system (prior and actual data), labeled $\mathcal{D}$. Such data can, e.g., be the set of all state transitions $\mathbf{s}, \mathbf{a}, \mathbf{s}_+$ observed so far, but also include some prior knowledge of the system. The strictest notion of safety hence becomes a probabilistic counterpart of (6), which can be defined in terms of constraint satisfaction as:

$$\sigma := \mathbb{P}\left[\, \mathbf{s}_i \in \mathcal{X} \quad \forall i \,|\, \mathcal{D} \,\right]. \tag{7}$$

Probability (7) is epistemological, and to be understood in the context of Bayesian hypothesis testing. It underlines that safety can only be a belief conditioned on our current knowledge of the system at hand. In this paper, we adopt that operational notion of knowledge-based safety and label a policy satisfying (7) as σ-safe. Assessing probability (7) in practice can be difficult, but it can arguably be done in several ways:

1. Direct data-based: data $\mathcal{D}$ are collected on the real system, and used to infer an estimation of σ, without using a model of (1). The obvious difficulty here lies in the need for collecting an extremely large data set if a policy with σ close to one is to be designed. Real data are costly, especially for safety-critical systems, and designing policies that achieve a high σ can be unrealistic in that context.
2. Direct model-based: a "pessimistic" simulation model of the real system is constructed (see Equations (8)-(11) below) from $\mathcal{D}$, and σ is estimated in silico via Monte Carlo methods. If the model is pessimistic, the in-silico estimation of σ converges (as more in-silico data are generated) to a lower bound for the true σ.
3. Indirect model-based: similarly to 2, a "pessimistic" control model of the real system is constructed, and used to build policies that are safe by construction for that model. Monte Carlo sampling is here replaced by an explicit or implicit propagation of the uncertainty set, based on an overestimation of the support of (1). The resulting σ is by construction a lower bound for (7).

Examples of approach 1. can be found in, e.g., [8,10]; examples of approach 2. can be found in [24,35]; and examples of approach 3. include, e.g., [6,20,29]. While approach 2. is often used in the context of safe reinforcement learning (providing only weak guarantees that hold upon convergence) and to certify control policies in practice, it is difficult to provide strong safety guarantees in that context. In this paper, we will consider the third approach, based on robust MPC techniques. Our intentions are two-fold. First, we present a theory that specifies formally how learning can be deployed in the MPC context without jeopardizing the closed-loop stability and safety of the resulting policy throughout the learning process. Second, we propose this framework as one viable approach to safe reinforcement learning if strong safety and stability guarantees are expected to be satisfied throughout the learning process.

The models of the real system used in approaches 2. and 3. above are often built as structured models that can, e.g., take the form:

$$\mathbf{s}_{i+1} = \mathbf{F}_{\boldsymbol{\vartheta}}\left(\mathbf{s}_i, \mathbf{a}_i, \mathbf{w}_i\right), \tag{8}$$

where $\mathbf{F}_{\boldsymbol{\vartheta}} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^n$, and $\boldsymbol{\vartheta}$ is the set of model parameters. Moreover, variable

$$\mathbf{w}_i \in \mathbb{W}_{\boldsymbol{\vartheta}} \subset \mathbb{R}^d, \tag{9}$$

is an external, typically stochastic disturbance contained in a compact set $\mathbb{W}_{\boldsymbol{\vartheta}}$, modeling the stochasticity of the real system. The notion of "pessimistic" model used above then requires that for any state-action pair $\mathbf{s}, \mathbf{a}$, the support of the real system dynamics density (1) is (almost entirely) included in the set:

$$\mathbb{D}_{\boldsymbol{\vartheta}}\left(\mathbf{s}, \mathbf{a}\right) = \left\{\, \mathbf{F}_{\boldsymbol{\vartheta}}\left(\mathbf{s}, \mathbf{a}, \mathbf{w}\right) \,|\, \forall\, \mathbf{w} \in \mathbb{W}_{\boldsymbol{\vartheta}} \right\}, \tag{10}$$

i.e.,

$$\int_{\mathbb{D}_{\boldsymbol{\vartheta}}(\mathbf{s},\mathbf{a})} \varphi\left[\, \mathbf{s}_+ \,|\, \mathbf{s}, \mathbf{a}\right] \mathrm{d}\mathbf{s}_+ = 1, \quad \forall\, \mathbf{s}, \mathbf{a}. \tag{11}$$

Condition (11) provides a formal definition of a "pessimistic" model, i.e., a model that includes the support of the real state transition (1). Note that this condition does not necessarily need to be conservative. Indeed (11) can in principle hold tightly, with $\mathbb{D}_{\boldsymbol{\vartheta}}$ covering the support of φ but not more. Condition (11) further entails that a policy guaranteeing that (6) is satisfied for all the possible trajectories resulting from (8)-(9) is safe by construction. Robust MPC techniques can be used to build such policies.

Similarly to (7), the validity of (11) is limited to our knowledge of the system supported by all data and prior knowledge available about it, i.e., $\mathcal{D}$. Condition (11) then ought to be regarded as probabilistic as well, in which case it becomes:

$$\tilde{\sigma} := \mathbb{P}\left[\,(11)\,|\,\mathcal{D}\,\right]. \tag{12}$$

One can easily verify that if a policy (5) ensures that the closed-loop trajectories of the model (8)-(11) respect the state constraints (6) at all time, then the probability that the policy is safe for the real system is at least $\tilde{\sigma}$, and $\sigma \geq \tilde{\sigma}$ holds. In practice, a minimum requirement for $\tilde{\sigma} > 0$ to hold is to ensure that:

$$\mathbf{s}_+ \in \mathbb{D}_{\boldsymbol{\vartheta}}\left(\mathbf{s},\mathbf{a}\right) \tag{13}$$

holds for all observed state transition triplets $(\mathbf{s},\mathbf{a},\mathbf{s}_+) \in \mathcal{D}$. Condition (13) then defines a set

$$\boldsymbol{\Theta}_{\mathcal{D}} := \left\{\,\boldsymbol{\vartheta}\;|\;\mathbf{s}_+ \in \mathbb{D}_{\boldsymbol{\vartheta}}\left(\mathbf{s},\mathbf{a}\right),\;\forall\;(\mathbf{s},\mathbf{a},\mathbf{s}_+) \in \mathcal{D}\,\right\}$$

where the model parameters $\boldsymbol{\vartheta}$ should be restricted, which is typically tackled via set-membership system identification methods [5]. In this paper we will consider a set $\boldsymbol{\Theta}_{\mathcal{D}}$ possibly formed from (13), and possibly further restricted by prior or structural knowledge of the system. For a complete discussion on the definition of $\boldsymbol{\Theta}_{\mathcal{D}}$ based on (13) and its deployment within safe RL, we refer to [31].

4 Robust MPC as a safe & stable policy

We will consider the use of robust MPC as a means to generate safe policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}_i\right)$ for system (1), subject to the associated performance index (4) and safety restriction (6), with the limitation (12). Note that these policies are parametrized by parameter $\boldsymbol{\theta}$, which defines the robust MPC scheme as we will discuss in the following. A strong point of robust MPC is that safety in the sense of Section 3 and stability can be enforced by construction. A non-trivial remaining question is then how to retain safety and stability through the learning process, where the parameters of the robust MPC scheme are regularly updated.

A generic robust MPC scheme is based on predicting the system evolution at future times based on actions given by a sequence of future, parametrized policies:

$$\mathbf{a}_k = \boldsymbol{\eta}^k_{\boldsymbol{\theta}}\left(\mathbf{v},\mathbf{s}_k\right), \tag{14}$$

where the indices k refer to future time instances, occurring at the corresponding physical times $i+k$ and $\mathbf{v}$ is a set of variables used to shape the policy sequence for the specific current state of the system $\mathbf{s}_i$. In this paper, we consider robust MPC schemes of the form:

$$\begin{aligned}
\hat{V}_{\boldsymbol{\theta}}(\mathbf{s}_i) = \min_{\mathbf{v}} \quad & \varphi_{\boldsymbol{\theta}}(\mathbf{v},\mathbf{s}_i) && \text{(15a)}\\
\text{s.t.} \quad & \mathbb{X}_{k+1} = \mathbf{F}_{\boldsymbol{\vartheta}}\left(\mathbb{X}_k, \boldsymbol{\eta}^k_{\boldsymbol{\theta}}\left(\mathbf{v},\mathbb{X}_k\right), \mathbb{W}_{\boldsymbol{\theta}}\right), && \text{(15b)}\\
& \mathbb{X}_0 = \mathbf{s}_i, \qquad \mathbb{X}^{\boldsymbol{\theta}}_N \subseteq \mathcal{X}^{\mathrm{f}}_{\boldsymbol{\theta}}, && \text{(15c)}\\
& \boldsymbol{\eta}^k_{\boldsymbol{\theta}}\left(\mathbf{v},\mathbb{X}_k\right) \subseteq \mathbb{U}, \qquad \mathbb{X}^{\boldsymbol{\theta}}_k \subseteq \mathcal{X}, && \text{(15d)}\\
& \forall\, k = 0,\ldots,N-1, && \text{(15e)}
\end{aligned}$$

where the state propagation (15b), called *tube*, is the extension of (8) to set propagation, i.e., we read (15b) as:

$$\mathbb{X}^{\boldsymbol{\theta}}_{k+1} = \left\{\,\mathbf{F}_{\boldsymbol{\vartheta}}\left(\mathbf{s}, \boldsymbol{\eta}^k_{\boldsymbol{\theta}}\left(\mathbf{v},\mathbf{s}\right), \mathbf{w}\right)\;|\,\forall\,\mathbf{s} \in \mathbb{X}^{\boldsymbol{\theta}}_k,\;\mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}\right\}. \tag{16}$$

The robust MPC scheme (15) defines a policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}\right)$ given by the first control input of the policy sequence adopted in the robust MPC scheme, i.e.:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\mathbf{s}_i\right) = \boldsymbol{\eta}^0_{\boldsymbol{\theta}}\left(\mathbf{v}^\star,\mathbf{s}_i\right), \tag{17}$$

where $\mathbf{v}^\star$ is the solution of (15). The robust MPC model used in (15b) is then an intrinsic and central component of the robust MPC scheme formulation. We will then consider that the model parameters $\boldsymbol{\vartheta}$ are part of the policy parameters $\boldsymbol{\theta}$, as they can be used to shape the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ delivered by the robust MPC scheme.

Set $\mathbb{U}$ in (15) represents the possible limitations on the feasible actions (e.g., actuator limitations), and the terminal set $\mathcal{X}^{\mathrm{f}}_{\boldsymbol{\theta}}$ must be constructed such that (15) being feasible entails that the state constraint $\mathbb{X}_k \subseteq \mathcal{X}$ can be enforced at all future times. This is typically achieved by resorting to a terminal control law which makes $\mathcal{X}^{\mathrm{f}}_{\boldsymbol{\theta}}$ forward invariant. The cost function (15a) is left unspecified here, as it can take different forms such as, e.g., a worst-case cost (as in min-max robust MPC); a nominal cost (as in tube MPC); an expected cost, or more elaborate risk-adverse costs. Function $\hat{V}_{\boldsymbol{\theta}}(\mathbf{s})$ receives an infinite value for the states $\mathbf{s}$ for which problem (15) is infeasible. For the sake of simplicity, we do not consider mixed state-input constraints here, although the proposed theory readily applies to that case.

We ought to stress that any Robust MPC formulation can be considered in our framework, e.g., robust MPC [20,6,32], possibly also combined with non-standard formulations such as [17]. Note that while nonlinear robust MPC is in general hard to formulate and solve, recent research has provided methods that can be applied in practice, see, e.g., [19,29,13]. Furthermore, it is arguably useful here to stress that the choice of using robust MPC to carry safe and stable policies, while not necessarily straightforward to implement, is nonetheless not restrictive. Indeed, the existence of a safe policy for the model (8)-(9) entails the existence of a robust MPC scheme (15) delivering a safe policy for the system model (8)-(9), and therefore a σ-safe policy for the real system (1). While these observations are well understood in the MPC community, we provide hereafter a Lemma that highlights their importance in a learning context.

Lemma 1 *Assume that there exists a feasible policy $\boldsymbol{\pi}$ such that the trajectories of the model* (8)-(9)*: (i) are in $\mathcal{X}$ with probability 1 for a set of initial conditions $\mathbf{s}_0 \in \mathcal{X}^0$; and (ii) are asymptotically stabilized to some set $\mathbb{L}$ with an associated Lyapunov function $\mathcal{V}_{\boldsymbol{\pi}}(\mathbf{s})$. Then there exists a robust MPC scheme of the form* (15) *that is σ-safe for the real system dynamics* (1)*, with probability at least σ of being stable for the real system dynamics* (1)*.*

Remark 1 *Before delivering the proof of this simple Lemma, it ought to be stressed here that its purpose is not to specify how the robust MPC scheme should be built, but rather to dismiss potential concerns that robust MPC is a limited tool for producing safe and stable policies. Indeed, it shows that if a safe and stable policy exists, then a robust MPC scheme that produces a safe and stable policy also exists.*

PROOF. For the first claim it is sufficient to prove that is it possible to setup a recursively feasible robust MPC scheme such that the model dynamics satisfy (6). Let us select $\boldsymbol{\eta}_{\boldsymbol{\theta}}^k$ as

$$\boldsymbol{\eta}_{\boldsymbol{\theta}}^k(\mathbf{v}, \mathbf{s}) = \boldsymbol{\pi}(\mathbf{s}) + \boldsymbol{\rho}_{\boldsymbol{\theta}}^k(\mathbf{v}, \mathbf{s}), \tag{18}$$

where $\boldsymbol{\rho}_{\boldsymbol{\theta}}^k$ is an arbitrary policy such that $\boldsymbol{\rho}_{\boldsymbol{\theta}}^k(0, \mathbf{s}) = 0$ for all $\mathbf{s}$, and select

$$\mathcal{X}_{\boldsymbol{\theta}}^{\mathrm{f}} = \mathcal{X}^0. \tag{19}$$

Then constraints (15b) and (15c) ensure that the robust MPC policy maintains the trajectories of the model (8)-(9) in the set $\mathcal{X}$ and that the robust MPC scheme is recursively feasible. Moreover, the choice of policy (18) ensures that the constraints are feasible because the trivial choice $\mathbf{v} = 0$ is feasible.

Concerning the second claim, we exploit the fact that the policy is stabilizing and there exists a Lyapunov function $\mathcal{V}_{\boldsymbol{\pi}}(\mathbf{s})$ associated with set $\mathbb{L}$. By selecting the cost as

$$\varphi_{\boldsymbol{\theta}}(\mathbf{v}, \mathbf{s}) = \mathcal{V}_{\boldsymbol{\pi}}(\mathbf{s}) + \mathbf{v}^\top \mathbf{v},$$

we obtain that $\boldsymbol{\eta}_{\boldsymbol{\theta}}^k(0, \mathbf{s})$ is optimal and the MPC value function is a Lyapunov function, i.e., it satisfies $\hat{V}_{\boldsymbol{\theta}}(\mathbf{s}) = \mathcal{V}_{\boldsymbol{\pi}}(\mathbf{s})$. Since policy $\boldsymbol{\pi}$ is safe and stabilizing by assumption, the corresponding robust MPC scheme is safe and stable for the model dynamics (8)-(9). Then, using (12), we conclude that the resulting policy is σ-safe for the real system dynamics (1) and stabilizing with probability σ. □

Lemma 1 is further discussed in Remark 2 below.

In practice, for simplicity, the parametrized policy (14) used in (15) is often selected as a nominal state-input reference sequence $\bar{\mathbf{u}}_{0,\dots,N-1}$, $\bar{\mathbf{x}}_{0,\dots,N-1}$ together with an additional linear state feedback, i.e.,

$$\boldsymbol{\eta}_{\boldsymbol{\theta}}^k(\mathbf{v}, \mathbf{s}) = \bar{\mathbf{u}}_k - K_{\boldsymbol{\theta}}(\mathbf{s} - \bar{\mathbf{x}}_k), \tag{20}$$

$$\mathbf{v} = \{\bar{\mathbf{u}}_{0,\dots,N-1}, \bar{\mathbf{x}}_{0,\dots,N-1}\}. \tag{21}$$

In that specific case, the policy (17) defined by the robust MPC scheme becomes

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{s}_i) = \boldsymbol{\eta}_{\boldsymbol{\theta}}^0(\mathbf{v}^\star, \mathbf{s}_i) = \bar{\mathbf{u}}_0^\star. \tag{22}$$

We define next some important sets in the space of parameters $\boldsymbol{\theta}$.

Definition 1 *Let us define the set $\boldsymbol{\Theta}_{\mathrm{F}}$ of parameters $\boldsymbol{\theta}$ such that the robust MPC scheme* (15) *is recursively feasible for the dynamics* (8)-(9) *for some non-empty set $\mathcal{X}_{\boldsymbol{\theta}}^0$ of initial conditions.*

Definition 2 *Let us define the set $\boldsymbol{\Theta}_{\mathrm{L}}$ of parameters $\boldsymbol{\theta}$ such that the value function $\hat{V}_{\boldsymbol{\theta}}(\mathbf{s})$ defined by* (15) *is a Lyapunov function for the dynamics* (8)-(9) *with respect to a set $\mathbb{L}_{\boldsymbol{\theta}}$.*

In order to make Definition 2 more concrete, we mention as an example the following conditions for $\hat{V}_{\boldsymbol{\theta}}(\mathbf{s})$ to be a Lyapunov function, which are classic in robust MPC:

(1) it is lower and upper bounded by $\mathcal{K}_\infty$ functions;
(2) $\hat{V}_{\boldsymbol{\theta}}(\mathbf{s}_+) \leq \gamma \hat{V}_{\boldsymbol{\theta}}(\mathbf{s}) + \delta_{\boldsymbol{\theta}}$ holds for all $\mathbf{s} \in \mathcal{X}_{\boldsymbol{\theta}}^0$ and where

$$\mathbf{s}_+ \in \{\mathbf{F}_{\boldsymbol{\vartheta}}(\mathbf{s}, \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{v}, \mathbf{s}), \mathbf{w}) \mid \forall \mathbf{w} \in \mathbb{W}_{\boldsymbol{\theta}}\},$$

for some positive constants $\delta_{\boldsymbol{\theta}}$, and $\gamma < 1$.

Then set $\mathbb{L}_{\boldsymbol{\theta}}$ is defined as:

$$\mathbb{L}_{\boldsymbol{\theta}} = \left\{ \mathbf{s} \,\middle|\, \hat{V}_{\boldsymbol{\theta}}(\mathbf{s}) \leq \frac{\delta_{\boldsymbol{\theta}}}{1 - \gamma} \right\}, \tag{23}$$

and the model (8)-(9) is stabilized to $\mathbb{L}_{\boldsymbol{\theta}}$ for any initial condition $\mathbf{s} \in \mathcal{X}_{\boldsymbol{\theta}}^0$, see [25]. In Section 6, we will use this definition of Lyapunov function for the sake of simplicity, though our results hold in general, *mutatis mutandis.*

In the context of Definitions 1 and 2, the σ-safety of the real system is guaranteed for $\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathcal{D}} \cap \boldsymbol{\Theta}_{\mathrm{F}}$, and the stability of the real system is guaranteed for $\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathcal{D}} \cap \boldsymbol{\Theta}_{\mathrm{L}}$ with probability at least σ. We note that establishing conditions on $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathrm{L}}$ is typically done in practice via min-max robust MPC or tube-based MPC [25]. We ought to stress that the conditions above are provided as an example which is fairly general, as it includes the case of [25, Section 3.4], as well as [20,32]. However, more generic conditions are applicable and can be used in our theory.

Remark 2 *We observe that Lemma 1 is of little practical use since it constructs a specific MPC scheme using a safe and stabilizing policy which is not known. Nevertheless, it allows us to obtain an important theoretical consideration. Provided that the MPC parametrization is rich enough, there does exist a parameter which yields a non-conservative model* (8)-(9)*, and a design of the MPC cost and feedback* (14) *such that the scheme we construct in the proof of the lemma can be captured by an adequate choice of the parameter. This means that there exists a parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathcal{D}} \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathrm{L}}$. Consequently, the intersection of these three sets is nonempty for a rich-enough parametrization of the robust MPC scheme.*

Assumption 1 *In the remainder of the paper, we will use the following assumptions:*

1. *The set of feasible initial conditions $\mathcal{X}^0_{\boldsymbol{\theta}}$ is compact and continuous in $\boldsymbol{\theta}$.*
2. *The conditional density* (1) *underlying the real system is bounded for all state-action pairs, i.e.,*

$$\varphi \left[\mathbf{s}_+ \,|\, \mathbf{s}, \mathbf{a} \right] \leq \bar{\varphi} < \infty, \quad \forall \mathbf{s}_+, \mathbf{s}, \mathbf{a}. \tag{24}$$

These technical assumptions will be required in some parts of the theory proposed in this paper. We ought to note here that the compactness of set $\mathcal{X}^0_{\boldsymbol{\theta}}$ is a mild assumption for a well-posed MPC scheme, and is standard in the robust MPC literature [6,20]. The continuity of the set with respect to $\boldsymbol{\theta}$ holds if, e.g., all functions involved in the MPC scheme are sufficiently differentiable, and if the MPC formulation satisfies standard regularity assumptions for Nonlinear Programs [22,7]. Finally, we stress that one specific setting of learning for robust MPC trivially satisfies this continuity assumption. Indeed, if the learning is not allowed to change the model and safety constraints in the robust MPC scheme, then $\mathcal{X}^0_{\boldsymbol{\theta}}$ is independent of $\boldsymbol{\theta}$ and continuity trivially follows. We should underline here that this case is in principle not restrictive. Indeed, as argued in [9], optimality can be recovered by adjusting the cost only, then such a choice does not introduce any theoretical restriction. Assumption 1.2. requires that all the states of the real system are subject to some stochasticity. It is a technical assumption aimed at simplifying the subsequent discussions, and could arguably be relaxed at the cost of making the subsequent argumentations significantly more technical.

4.1 Safety & Stability Constraints in RL

In the following, we will consider the safety and stability conditions detailed in this section as constraints applied to the RL steps updating the policy parameters. More specifically, similarly to [31], we consider that the RL steps taken on the robust MPC scheme are feasible steps $\Delta\boldsymbol{\theta}$ taken on the constrained optimization problem:

$$\min_{\boldsymbol{\theta}} \quad J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right), \tag{25a}$$

$$\text{s.t.} \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathrm{L}} \ \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}, \tag{25b}$$

in the sense that each parameter update $\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p + \Delta\boldsymbol{\theta}$, labelled by the index p, satisfies (25b) and reduces the cost (25a). More specifically, the RL steps will be computed according to:

$$\min_{\boldsymbol{\theta}_{p+1}} \quad \frac{1}{2}\left\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\right\|_H^2 + \alpha \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}_p}\right)^\top \left(\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\right), \tag{26a}$$

$$\text{s.t.} \quad \boldsymbol{\theta}_{p+1} \in \boldsymbol{\Theta}_{\mathrm{L}} \ \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}, \tag{26b}$$

for some positive-definite matrix $H \approx \nabla^2_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right)$, and some $\alpha \in (0, 1]$. We should recall here that for any $H \succ 0$ and α small enough, the sequence $\boldsymbol{\theta}_{0,\ldots,\infty}$ stemming from (26) converges to a (possibly local) solution of (25) [22].

It is useful to observe here that classic policy gradient methods [26,28] in RL are typically based on (26a), with the exclusion of the safety and stability constraint (26b). The identity matrix is often used for H when the policy is based on very high dimensional approximators such as, e.g., Deep Neural Networks.

We will assume here that the gradient $\nabla_{\boldsymbol{\theta}} J$ in (26a) is either evaluated directly via actor-critic or policy search techniques, or replaced by a surrogate based on Q-learning techniques, all formed using data collected on the real system in closed-loop with policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. The safety and stability constraints (25b) will then be built based on (6), (8), (9), and (13). In the remainder of the paper, a mild technical assumption on the Nonlinear Program (26) will be very helpful.

Assumption 2 *The solution $\boldsymbol{\theta}_{p+1}$ of* (26) *is continuous with respect to α in a neighborhood of $\alpha = 0$.*

Assumption 2 follows from technical assumptions on the set $\boldsymbol{\Theta}_{\mathrm{L}} \ \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$, which we propose to not discuss extensively here for the sake of brevity. In particular, we note that Assumption 2 naturally holds if the set $\boldsymbol{\Theta}_{\mathrm{L}} \ \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$ can be represented by a finite set of continuous inequality constraints, and if the resulting problem (26) fulfils classical regularity assumptions and sufficient second-order conditions (SOSC) [22].

An important question that needs to be addressed is how feasible parameter updates $\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p + \Delta\boldsymbol{\theta}$ resulting from (26) can be implemented in the robust MPC scheme without jeopardizing the safety and stability of the closed-loop system. We discuss the safety question in the next section using two different approaches.

5 Recursive Feasibility with RL-Based Parameter Updates

Let us consider a sequence of parameters $\boldsymbol{\theta}_{0,\ldots,\infty}$ resulting from (26), and consider that each parameter $\boldsymbol{\theta}_p$ of that sequence is applied for a certain amount of time (i.e., at least one sampling time of the robust MPC scheme). This section provides conditions such that this sequence of parameter updates does not jeopardize the safety of the corresponding sequence of policies $\boldsymbol{\pi}_{\boldsymbol{\theta}_{0,\ldots,\infty}}$ resulting from the corresponding robust MPC schemes. Note that a parameter update $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$ occurring at a time sample i means here that $\mathbf{a}_i = \boldsymbol{\pi}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_i)$ and the inputs $\mathbf{a}_j = \boldsymbol{\pi}_{\boldsymbol{\theta}_p}(\mathbf{s}_j)$ with $j < i$ are used from the previous parameter update. The next theorem provides a first set of conditions for ensuring the safety of the parameter updates.

Theorem 1 *Assume that for all p, parameter $\boldsymbol{\theta}_p$ satisfies $\boldsymbol{\theta}_p \in \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$, and that the initial conditions $\mathbf{s}_0$ are in the set $\mathcal{X}^0_{\boldsymbol{\theta}_0}$. If each parameter update $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$ takes place in a state $\mathbf{s}_i$ such that*

$$\mathbf{s}_i \in \mathcal{X}^0_{\boldsymbol{\theta}_{p+1}} \tag{27}$$

holds, then the closed-loop trajectories $\mathbf{s}_{0,\ldots,\infty}$ resulting from applying the sequence of policies $\boldsymbol{\pi}_{\boldsymbol{\theta}_{0,\ldots,\infty}}$ is σ-safe.

PROOF. Assuming that (11) holds, then a standard result for robust MPC is that if $\boldsymbol{\theta}_{p+1} \in \boldsymbol{\Theta}_{\mathrm{F}}$ and if the initial state at which policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_{p+1}}$ is deployed satisfies condition (27), then policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_{p+1}}$ ensures that the state trajectories are feasible at all time with unitary probability. We then observe that if every parameter update $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$ is applied under condition (27), then each policy keeps the state trajectories feasible. As a result, if (27) is ensured at every parameter update $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$, then the entire state trajectory $\mathbf{s}_{0,\ldots,\infty}$ resulting from the policy sequence $\boldsymbol{\pi}_{\boldsymbol{\theta}_{0,\ldots,\infty}}$ remains feasible at all time.

If statement (11) does not hold, then the closed-loop trajectories $\mathbf{s}_{0,\ldots,\infty}$ may become infeasible, though not necessarily. Hence if statement (11) holds with a probability σ, then the closed-loop trajectories $\mathbf{s}_{0,\ldots,\infty}$ have probability no smaller than σ to be feasible. □

The results of Theorem 1 can be leveraged in practice by solving the robust MPC schemes associated to both $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_{p+1}$ in parallel at every sampling instant and selecting the control input associated to $\boldsymbol{\theta}_{p+1}$ as soon as the MPC scheme associated to $\boldsymbol{\theta}_{p+1}$ is feasible.

The theorem ensures the recursive feasibility of the sequence of robust MPC schemes such that the closed-loop state trajectories are contained in the set $\mathcal{X}$, within the framework presented in Section 3. An important caveat, though, is that there is no guarantee that condition (27) can be met in finite time by the closed-loop trajectories under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_p}$. As a result, it might be possible that a parameter update $\boldsymbol{\theta}_{p+1}$, though feasible for (25), yields an update condition (27) that never becomes satisfied, hence blocking the learning process. The remainder of this section proposes two different approaches to tackle that issue, using either backtracking or additional constraints in (25).

In order to support and simplify the coming argumentation, it is useful to introduce a technical lemma. Let us consider the trivial locally compact measure on $\mathbb{R}^n$, associating to any compact set $\mathbb{A} \subset \mathbb{R}^n$ the bounded positive real number:

$$\mu(\mathbb{A}) = \int_{\mathbb{A}} \mathrm{d}\mathbf{s}. \tag{28}$$

Lemma 2 *Suppose that Assumption 1.2 holds and consider an arbitrary policy $\mathbf{a} = \boldsymbol{\pi}(\mathbf{s})$, yielding a (continuous) Markov Chain $\mathbf{s}_{0,\ldots,\infty}$. Then for any set $\mathbb{A}$ and initial condition $\mathbf{s}_0 \in \mathbb{A}$, the following inequality holds:*

$$\mathbb{P}[\,\mathbf{s}_{0,\ldots,\infty} \in \mathbb{A}\,] \leq \mu(\mathbb{A})\,\bar{\varphi} \tag{29}$$

where $\bar{\varphi}$ is defined by (24).

PROOF. Denoting $\phi[\mathbf{s}_k]$ the density of the Markov chain at time k, we observe that:

$$\phi[\mathbf{s}_k] := \int \varphi\left[\,\mathbf{s}_k \,|\, \mathbf{s}_{k-1}, \boldsymbol{\pi}(\mathbf{s}_{k-1})\,\right] \phi[\mathbf{s}_{k-1}] \mathrm{d}\mathbf{s}_{k-1} \leq \bar{\varphi} \tag{30}$$

holds for any $k > 0$. It follows that

$$\mathbb{P}[\mathbf{s}_k \in \mathbb{A}] = \int_{\mathbb{A}} \phi[\mathbf{s}_k] \mathrm{d}\mathbf{s}_k \leq \mu(\mathbb{A})\,\bar{\varphi}. \tag{31}$$

Then (29) follows from the Fréchet inequalities, stating:

$$\mathbb{P}[\mathbf{s}_{0,\ldots,\infty} \in \mathbb{A}] \leq \inf_k \; \mathbb{P}[\mathbf{s}_k \in \mathbb{A}] \leq \mu(\mathbb{A})\,\bar{\varphi}. \tag{32}$$

□

5.1 Parameter Update via Backtracking

In this subsection, we consider the use of backtracking on the parameter updates computed according to (26) to ensure the feasibility of updating the parameters in finite time. For the sake of simplicity in the following developments, rather than a line-search strategy [22], we use a gradient adaptation strategy in the cost (26a), by iteratively reducing parameter α, therefore generating a step ranging from a full step ($\alpha = 1$) to $\Delta\boldsymbol{\theta} = 0$ (with $\alpha = 0$). The following theorem then guarantees that there is some $\alpha > 0$ such that the probability that the parameter update condition (27) is not met in finite time is less than $1 - \sigma$.

Theorem 2 *Consider the closed loop trajectory* $\mathbf{s}_{i,\dots,\infty}$ *under policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}_p}$ *starting at the physical sampling time* i *with the initial state* $\mathbf{s}_i$. *Consider the parameter update* $\boldsymbol{\theta}_{p+1}(\alpha)$ *(where we highlight the dependency on* α*) resulting from* (26) *and suppose that Assumptions 1-2 hold. Then the probability that the update condition* (27) *is not met in finite time can be made arbitrarily small by selecting* α *small enough, i.e., the following limit holds:*

$$\lim_{\alpha\to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{p+1}(\alpha)}\right] \leq 1 - \sigma. \tag{33}$$

A simple interpretation of Theorem 2 is that it is always possible to backtrack to a short-enough parameter update (α small enough) such that the update condition (27) becomes satisfied in finite time with probability arbitrarily close to $1 - \sigma$.

The intuition behind this result is that, by continuity arguments, $\mathcal{X}^0_{\boldsymbol{\theta}_{p+1}(\alpha)}$ tends to $\mathcal{X}^0_{\boldsymbol{\theta}_p}$ as α becomes small, such that the two sets match asymptotically. Moreover, we observe that under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_p}$, the closed-loop trajectories evolve in set $\mathcal{X}^0_{\boldsymbol{\theta}_p}$ and the update is infeasible if they are outside of set $\mathcal{X}^0_{\boldsymbol{\theta}_{p+1}(\alpha)}$. It follows that for a parameter update to be blocked forever despite α being arbitrarily small, if (11) holds, the closed-loop state trajectories under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_p}$ need to evolve on an infinitely small set. This would require unbounded densities in the real closed-loop dynamics (1), which is excluded by Assumption 1, or that (11) does not hold. We formalize these explanations in the next proof.

PROOF. (of Theorem 2) If (11) holds, we first observe that $\boldsymbol{\theta}_{p+1}(0) = \boldsymbol{\theta}_p$ trivially holds, such that

$$\mathbf{s}_{i,\dots,\infty} \in \mathcal{X}^0_{\boldsymbol{\theta}_{p+1}(0)} \tag{34}$$

holds by construction. Let us further define the set:

$$\Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_{p+1}) = \left\{ \mathbf{s} \;\middle|\; \mathbf{s} \in \mathcal{X}^0_{\boldsymbol{\theta}_p} \quad \text{and} \quad \mathbf{s} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{p+1}} \right\}, \tag{35}$$

such that $\Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_p) = \emptyset$. A trajectory $\mathbf{s}_{k,\dots,\infty}$ in closed-loop under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_p}$ that never satisfies the update condition (27) must evolve in $\Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_{p+1})$. We observe that by Assumption 2 $\boldsymbol{\theta}_{p+1}(\alpha)$ is continuous in a neighborhood of $\alpha = 0$, and by Assumption 1.1 we have that the set $\Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_{p+1})$ is continuous in $\boldsymbol{\theta}_{p+1}$. It follows that

$$\lim_{\alpha\to 0} \mu\left(\Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_{p+1}(\alpha))\right) = 0. \tag{36}$$

We can then conclude using Lemma 2:

$$\begin{aligned} &\lim_{\alpha\to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{p+1}(\alpha)}\right] = \\ &\quad \lim_{\alpha\to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \in \Delta\mathcal{X}^0(\boldsymbol{\theta}_p, \boldsymbol{\theta}_{p+1}(\alpha))\right] = 0. \end{aligned} \tag{37}$$

Since (11) holds with probability σ, (33) readily follows. □

Algorithm 1: Safe and Stable learning - backtracking

```
   Input: MPC parameter $\boldsymbol{\theta}$, and $\varrho$, $n$, $H$
1  while Learning do
2      Set update = false, $\alpha = 1$, fail = 0
3      Compute $\boldsymbol{\theta}_+$ from (26)
4      while not update do
5          Compute MPC solution $\mathbf{u}_0$ from $\boldsymbol{\theta}$
6          Compute MPC solution $\mathbf{u}_0^+$ from $\boldsymbol{\theta}_+$
7          if MPC solution from $\boldsymbol{\theta}_+$ is feasible then
8              Set $\mathbf{u}_0 \leftarrow \mathbf{u}_0^+$ and $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_+$
9              update = true
10         else
11             fail = fail + 1
12         Apply input $\mathbf{u}_0$ to the system
13         if fail $\geq n$ then
14             $\alpha = \varrho\alpha$ and recompute $\boldsymbol{\theta}_+$ from (26)
```

A practical implementation of the backtracking approach is detailed in Algorithm 1. The implementation consists in reducing parameter α if the update condition is not met for n time steps. We ought to stress here that lines 3 and 14 of Algorithm 1 can be performed offline independently of the state of the system and of the robust MPC schemes. It follows that the online computational burden is limited to solving two independent robust MPC schemes, possibly in parallel.

5.2 Parameter Updates via Constrained Feasibility

As an alternative to backtracking, we propose next an approach imposing additional constraints in (26). We then no longer rely on taking short-enough steps in $\boldsymbol{\theta}$ to achieve the feasibility of the parameter updates, but rather form an update that is feasible by construction. This entails that updates can be performed at every time step i such that $p = i$, hence we will use the notation $\boldsymbol{\theta}_i$ throughout this section.

We define as $\mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i))$ the 1-step dispersion set starting from state $\mathbf{s}_i$, applying the input $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)$, and using set $\mathbb{W}_{\boldsymbol{\theta}_{i+1}}$ to model the stochasticity of the system. Note the subtle but important difference with $\mathbb{X}_1^{\boldsymbol{\theta}_{i+1}} = \mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i))$, where we apply action $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i)$ instead of $\mathbf{a} = \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)$.

Theorem 3 *The parameter update $\boldsymbol{\theta}_{i+1}$ given by*

$$\min_{\boldsymbol{\theta}_{i+1}} \quad \frac{1}{2}\left\|\boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i\right\|_H^2 + \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}\right)^\top \left(\boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i\right), \tag{38a}$$

$$\text{s.t.} \quad \boldsymbol{\theta}_{i+1} \in \boldsymbol{\Theta}_{\mathrm{L}} \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}, \tag{38b}$$

$$\mathcal{X}^0_{\boldsymbol{\theta}_{i+1}} \supseteq \mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)). \tag{38c}$$

satisfies (27) *by construction with probability at least σ.*

PROOF. Equation (11) entails $\mathbf{s}_{i+1} \in \mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i))$. Using (38c), $\mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)) \subseteq \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}$ holds, and robust MPC is feasible for all possible realizations of $\mathbf{s}_{i+1}$, i.e., $\mathbf{s}_{i+1} \in \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}$. Since (11) holds with probability σ, (27) holds with probability at least σ. □

We elaborate next on how constraint (38c) can be formulated. We observe that recursive feasibility of MPC (15) implies that, if a given state is feasible, then the tube around the predicted trajectory is also feasible, such that

$$\mathbb{X}_1^{\boldsymbol{\theta}_{i+1}}(\mathbf{s}_i, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)) \subseteq \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}} \quad \Leftarrow \quad \mathbf{s}_i \in \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)), \tag{39}$$

where $\mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}(\mathbf{a})$ defines the set of states $\mathbf{s}$ for which the robust MPC scheme (15) is feasible under the additional constraint $\bar{\mathbf{u}}_0 = \mathbf{a}$. The second condition in (39) is more easily written as a condition on the parameters and the nominal MPC trajectory: a detailed discussion on how this is done is provided in [31] for linear tube MPC. The main difference between that approach and the one used in this paper is that in that case the constraint takes the form $\mathbf{h}(\mathbf{v}, \boldsymbol{\theta}) \leq 0$, while in this paper it takes the form $\mathbf{h}(\mathbf{v}^\star(\boldsymbol{\theta}), \boldsymbol{\theta}) \leq 0$. Since the main idea is unchanged, we do not provide further details for the sake of brevity.

We prove next that constraint (38) is non-blocking, i.e., that the parameter update yielded by (38) cannot be $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i$ at all times, unless $\boldsymbol{\theta}_i = \boldsymbol{\theta}_\star$. Note that Theorem 3 does not require Assumption 1 nor Assumption 2. However, both assumptions are needed in order to be able to prove the non-blocking property.

Theorem 4 *Consider the closed loop trajectory $\mathbf{s}_{i,\dots,\infty}$ under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_i}$ starting at the physical sampling time i with the initial state $\mathbf{s}_i$. Suppose that Assumptions 1-2 hold. Assume further that Assumption 1.1 holds also for $\mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}(\mathbf{a})$, i.e., if the first action is fixed in the MPC scheme, the set of feasible initial conditions is compact and continuous in $\boldsymbol{\theta}$. Then, the probability that* (38) *yields $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i \neq \boldsymbol{\theta}_\star$ for all times is zero, i.e.,*

$$\mathbb{P}\left[\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p \neq \boldsymbol{\theta}_\star, \; p = i, \dots, \infty\right] = 0. \tag{40}$$

PROOF. In order to prove the result, we will prove that there does exist a parameter update $\boldsymbol{\theta}_{i+1} \neq \boldsymbol{\theta}_i$ which does decrease the cost (38a) and satisfy (38c). To that end, we consider $\boldsymbol{\theta}_{i+1}(\alpha)$ resulting from (26), and we prove next that

$$\begin{aligned} &\lim_{\alpha \to 0} \mathbb{P}\left[\mathcal{X}^0_{\boldsymbol{\theta}_{i+1}(\alpha)} \not\supseteq \mathbb{X}_1(\mathbf{s}_j, \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_j)), \; j = i, \dots, \infty\right] \\ &\qquad \leq \lim_{\alpha \to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}(\alpha)}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)))\right] = 0, \end{aligned} \tag{41}$$

where we used (39) to obtain the inequality above. Because $H \succ 0$, any update $\boldsymbol{\theta}_{i+1}(\alpha)$ reducing (26a) must also be a descent direction for (38a). Consequently, if additionally $\boldsymbol{\theta}_{i+1}(\alpha)$ is feasible for (38c), then (38) cannot yield a 0 update.

Since by using $\alpha = 0$, (26) yields $\boldsymbol{\theta}_{i+1}(0) = \boldsymbol{\theta}_i$, we have

$$\mathbf{s}_{i,\dots,\infty} \in \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}(0)}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)) = \mathcal{X}^0_{\boldsymbol{\theta}_i}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)) = \mathcal{X}^0_{\boldsymbol{\theta}_i}.$$

We use (39) to define the set:

$$\begin{aligned} &\Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}) \\ &\qquad = \left\{ \mathbf{s} \;\middle|\; \mathbf{s} \in \mathcal{X}^0_{\boldsymbol{\theta}_i}, \text{ and } \mathbf{s} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)) \right\}, \end{aligned} \tag{42}$$

i.e., the set for which $\boldsymbol{\theta}_{i+1}$ violates the constraint (38c). Note that $\Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) = \emptyset$. Consider a trajectory $\mathbf{s}_{i,\dots,\infty}$ in closed-loop under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_i}$ such that parameter $\boldsymbol{\theta}_{i+1}(\alpha)$ solves (26) but never satisfies constraint (38c). By definition such trajectory must evolve in set $\Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}(\alpha))$. We observe that, by Assumption 2, $\boldsymbol{\theta}_{i+1}(\alpha)$ is continuous in a neighborhood of $\alpha = 0$, and by assumption we have that the set $\Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1})$ is continuous in $\boldsymbol{\theta}_{i+1}$. It follows that

$$\lim_{\alpha \to 0} \mu\left(\Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}(\alpha))\right) = 0. \tag{43}$$

We can then conclude using Lemma 2:

$$\begin{aligned} &\lim_{\alpha \to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \notin \mathcal{X}^0_{\boldsymbol{\theta}_{i+1}(\alpha)}(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i))\right] = \\ &\qquad \lim_{\alpha \to 0} \mathbb{P}\left[\mathbf{s}_{i,\dots,\infty} \in \Delta\mathcal{X}^0_{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\mathbf{s}_i)}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}(\alpha))\right] = 0. \end{aligned} \tag{44}$$

□

6 Stability of MPC with Parameter Updates

In the previous section, we investigated the recursive feasibility of performing RL-based parameter updates on the MPC scheme. In this section we will discuss the stability of a sequence of MPC schemes satisfying the recursive feasibility conditions discussed above. We will first discuss the joint stability of the state $\mathbf{s}$ and parameters $\boldsymbol{\theta}$; and then relax our assumptions, treat the updates of $\boldsymbol{\theta}$ as a perturbation acting on the system, and prove Input-to-State Stability (ISS).

In order to discuss joint state and parameter stability, we will show that, assuming that the sequence of parameters converges linearly, the sequence of MPC policies stabilizes the system in the state-parameter space. If $\boldsymbol{\theta}_p \in \boldsymbol{\Theta}_{\mathrm{L}} \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$ for all p, the sequence of parameters $\boldsymbol{\theta}_{0,\ldots,\infty}$ yields a sequence of Lyapunov functions $\hat{V}_{\boldsymbol{\theta}_p}$ on their respective feasible sets $\mathcal{X}^0_{\boldsymbol{\theta}_p}$. Hence each MPC with parameter $\boldsymbol{\theta}_p$ is stabilizing the system trajectory to the corresponding level set $\mathbb{L}_{\boldsymbol{\theta}_p}$. The stability of the system trajectories when updating the parameters $\boldsymbol{\theta}_p$ can then be investigated by piecing together the individual Lyapunov functions $\hat{V}_{\boldsymbol{\theta}_{0,\ldots,\infty}}$, and by assuming some regularity condition on the functions $\hat{V}_{\boldsymbol{\theta}_{0,\ldots,\infty}}$ as well as a sufficiently fast convergence of the parameter sequence $\boldsymbol{\theta}_{0,\ldots,\infty}$. This statement is formalized in the next theorem, for which we need the two following assumptions.

Assumption 3 *At every parameter update $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$, the inequality:*

$$\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_\star\| \leq r\,\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\| \tag{45}$$

holds for some $r \in]0,1[$, where $\boldsymbol{\theta}_\star$ is the solution of (25).

Assumption 4 *The condition*

$$\sup_{\mathbf{s}\in\mathcal{X}^0_{\boldsymbol{\theta}_p}\cap\mathcal{X}^0_{\boldsymbol{\theta}_{p+1}}} \left|\hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}) - \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s})\right| \leq \alpha_V\,\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\| \tag{46}$$

holds for all p for some $\alpha_V > 0$.

Before providing the result, let us discuss these two assumptions. Assumption 3 essentially requires that the parameter updates converge at a linear rate. This type of convergence holds for exact gradient methods, i.e., if the policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ used to compute the parameter updates in, e.g., (26) is evaluated exactly. We ought to stress here that in a learning context, the gradient is typically evaluated from data and therefore is stochastic by nature. The amount of stochastic perturbation around the average value asymptotically decays as the amount of data used in the gradient evaluation increases. As a result, Assumption 3 is to be taken as asymptotically valid, as well as the subsequent result presented in Theorem 5. A stability result requiring an assumption weaker than Assumption 3 is presented later in the text. That latter result is, however, weaker.

In order to discuss the conservatism of the regularity Assumption 4, we observe first that in general both the policy and the value function can be discontinuous [25]. Indeed, the continuity of the value function of optimization problems is known to be intricate, with the notable exception of linear robust MPC with a quadratic cost and polyhedral uncertainty. Assuming that the parameter sequence $\boldsymbol{\theta}_{0,\ldots,\infty}$ belongs entirely to a bounded and connected (BC) set Θ such that $\mathcal{X}^0_{\boldsymbol{\theta}}$ is non-empty everywhere in Θ, we observe that (46) holds if $\hat{V}_{\boldsymbol{\theta}}$ is Lipschitz continuous on Θ with a bounded Lipschitz constant for all $\mathbf{s}$ in the applicable domain of definition. This, in turn, holds if for all $\mathbf{s}$ in the domain of definition, $\hat{V}_{\boldsymbol{\theta}}$ is almost everywhere differentiable with respect to $\boldsymbol{\theta}$ on Θ, with bounded and Lebesgue integrable derivatives. Since in the general case discontinuities cannot be excluded, one might be tempted to conclude that (46) is in general violated. However, one should consider that, under very mild regularity assumptions, such discontinuities can only occur on a set of zero measure, i.e., condition (46) holds for almost all $\mathbf{s}$. Indeed, whenever the optimal MPC solution satisfies the strong second order sufficient conditions for optimality, a suitable constraint qualification, and strict complementarity holds, then both the policy and the value function are continuous and differentiable with respect to the parameter $\boldsymbol{\theta}$ [22,7]. In case strict complementarity does not hold, then directional derivatives do exist and continuity is not lost, such that (46) holds. A typical situation in which the policy and eventually also the value function can become discontinuous is when one of the two other conditions is not met. We observe that essentially all NMPC solvers require these two conditions to hold, such that one can conclude that any state-parameter combination which does not pose issues for NMPC solvers satisfies (46). Since the set of problematic points is of zero measure for well-posed problems, the probability of visiting such a state-parameter pair is zero for non-pathological systems (1).

Theorem 5 *Suppose that Assumptions 3-4 hold. Let us additionally assume that each parameter is given by* (26) *or* (38) *such that $\boldsymbol{\theta}_p \in \boldsymbol{\Theta}_{\mathrm{L}} \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$, and the parameter updates satisfy the parameter update conditions* (27) *or* (38c)*. Then the sequence of robust MPC schemes with parameters $\boldsymbol{\theta}_{0,\ldots,\infty}$ is asymptotically stable in the joint state-parameter update space for any $\mathbf{s}_0 \in \mathbb{X}^0_{\boldsymbol{\theta}_0}$, and steers the system trajectory to the level set $\mathbb{L}_{\boldsymbol{\theta}_\infty}$.*

PROOF. Consider an augmentation of the state $\mathbf{s}$ with the current parameters $\boldsymbol{\theta}_p$. Let us label the augmented state $\mathbf{S}$. We then propose the candidate Lyapunov function:

$$W(\mathbf{S}) = \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}) + \zeta\Delta_p \tag{47}$$

where we label $\Delta_p := \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\|$, and ζ is a positive constant. Function (47) tackles the regular state space of the system jointly with the parameter update space, and will allow us to establish stability in that joint space for ζ large enough. We first observe that since $\hat{V}_{\boldsymbol{\theta}_{0,\ldots,\infty}}(\mathbf{s})$ are Lyapunov functions, for any $\boldsymbol{\theta}_p$, $W(\mathbf{S})$ is a Lyapunov function in between parameter updates, i.e., W is decreasing along the system trajectory. Moreover, since Δ_p is a norm in the state-parameter space, $W(\mathbf{S})$ is adequately lower and upper bounded if $\hat{V}_{\boldsymbol{\theta}_p}$ is. Finally, since by assumption $\Delta_p \to 0$, the system state $\mathbf{s}$ is eventually steered towards $\mathbb{L}_{\boldsymbol{\theta}_\infty}$. We observe that between parameter updates the decrease of $W(\mathbf{S})$ under a specific parameter $\boldsymbol{\theta}_p$ holds from:

$$\begin{aligned} W(\mathbf{S}_{i+1}) &= \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_{i+1}) + \zeta\Delta_p & (48)\\ &\leq \gamma\hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_i) + \delta_{\boldsymbol{\theta}_p} + \zeta\Delta_p < W(\mathbf{S}_i), & (49)\end{aligned}$$

for any $\mathbf{s}_i$ outside of $\mathbb{L}_{\boldsymbol{\theta}_p}$.

Upon updating the parameter $\boldsymbol{\theta}_p \to \boldsymbol{\theta}_{p+1}$ at a specific time instant i, we observe that:

$$\begin{aligned} W(\mathbf{S}_{i+1}) - W(\mathbf{S}_i) = &\hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_{i+1}) - \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_i) \\ &+ \zeta\left(\Delta_{p+1} - \Delta_p\right). \end{aligned} \tag{50}$$

If the state at time $\mathbf{s}_i$ lies outside of the level set $\mathbb{L}_{\boldsymbol{\theta}_{p+1}}$ such that:

$$\hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_{i+1}) < \hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_i) \tag{51}$$

holds under the MPC with parameter $\boldsymbol{\theta}_{p+1}$, then W is decreasing over the parameter updates at time i if:

$$\begin{aligned} &\hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_{i+1}) - \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_i) + \zeta\left(\Delta_{p+1} - \Delta_p\right) < \\ &\quad \hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_i) - \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_i) + \zeta\left(\Delta_{p+1} - \Delta_p\right) \leq 0. \end{aligned} \tag{52}$$

Using (46), condition (52) holds if:

$$\begin{aligned} &\hat{V}_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_i) - \hat{V}_{\boldsymbol{\theta}_p}(\mathbf{s}_i) + \zeta\left(\Delta_{p+1} - \Delta_p\right) \leq \\ &\alpha_V\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\| + \zeta\left(\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_\star\| - \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\|\right) \leq 0. \end{aligned} \tag{53}$$

Using (45) we observe that

$$\begin{aligned} \|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\| &\leq \|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_\star\| + \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\| & (54)\\ &\leq (r+1)\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\|. & (55)\end{aligned}$$

It follows that (53) holds if

$$\begin{aligned} &\alpha_V\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_p\| + \zeta\left(\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_\star\| - \|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\|\right) \leq \\ &\quad \leq \alpha_V(r+1)\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\| + \zeta(r-1)\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\| \leq 0. \end{aligned} \tag{56}$$

Hence W decreases when $\mathbf{s}_i$ lies outside of the level set $\mathbb{L}_{\boldsymbol{\theta}_{p+1}}$ if:

$$\alpha_V(r+1) + \zeta(r-1) \leq 0, \tag{57}$$

which can always be ensured by choosing:

$$\zeta \geq \frac{\alpha_V(r+1)}{1-r}. \tag{58}$$

As a result at all time k whether a parameter update takes place or not, either function W is decreasing or the system trajectory is contained in the level set $\mathbb{L}_{\boldsymbol{\theta}_p}$ corresponding to the MPC parameters in use. □

We elaborate in the next remarks on the assumptions and the stability claim made by Theorem 5.

Remark 3 *The decrease of function W at a time instant i with corresponding parameter index p is ensured for any $\mathbf{s}_i$ outside the level sets $\mathbb{L}_{\boldsymbol{\theta}_p}$, regardless of whether a parameter update has occurred or not. However, if $\mathbf{s}_i \in \mathbb{L}_{\boldsymbol{\theta}_p}$, then W is not guaranteed to decrease, but the trajectory $\mathbf{s}_{i,i+1,\ldots}$ is guaranteed to remain in the level set $\mathbb{L}_{\boldsymbol{\theta}_p}$ until the next parameter update occurs.*

Theorem 5 guarantees the stabilization of the system trajectory in the state-parameter space, and under the Lyapunov function W, to the sequence of level sets $\mathbb{L}_{\boldsymbol{\theta}_p}$ converging to $\mathbb{L}_{\boldsymbol{\theta}_\infty}$. Theorem 5 hence guarantees the stability of the system trajectory under fast parameter updates (e.g., at every sampling time i), albeit the parameter updates entering in the Lyapunov function via the norm Δ_p can temporarily drive the system trajectory away from the level sets (due to the fact that stability is guaranteed in the state-parameter update space, as opposed to the state space alone).

Hence a case covered by Theorem 5 and expected in practice is one where the parameter updates are fairly slow and small compared to the system dynamics, possibly yielding a situation where the system trajectory is stabilized to $\mathbb{L}_{\boldsymbol{\theta}_\infty}$ mostly by moving from level set to level set, i.e., $\mathbb{L}_{\boldsymbol{\theta}_p} \to \mathbb{L}_{\boldsymbol{\theta}_{p+1}} \to \ldots$, without W systematically decreasing. Theorem 5, however, guarantees that updating the parameters faster and more aggressively does not jeopardize the system stability.

Remark 4 *Let us further comment on Assumption 3, i.e., convergence of the parameters sequence $\boldsymbol{\theta}_{0,\ldots,\infty}$ delivered by the RL scheme. As previously discussed, many*

RL methods deliver a sequence of parameters that is stochastic by nature, because they are based on measurements taken from a stochastic system. We then observe that Equation (45) *is only satisfied asymptotically for large data sets. For RL methods based on very small data sets, such as, e.g., to the extreme those using basic stochastic gradient methods, one could consider an extension of Theorem 5 where the decrease of W holds only in a stochastic sense, or as practical stability or ISS. To that end, one can, e.g., assume*

$$\|\boldsymbol{\theta}_{p+1} - \boldsymbol{\theta}_\star\| \le r\,\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_\star\| + q, \tag{59}$$

which implies that the parameter updates converge to a neighborhood of zero. The main conclusions of the Theorem still hold, mutatis mutandis, with asymptotic stability replaced by practical stability. A formal discussion of this extension is not provided here for the sake of brevity, and we rather discuss next an alternative relaxation of Assumptions 3-4.

While the result of Theorem 5 is strong, it also requires some assumptions that do not necessarily hold, and is typically only asymptotically valid. We provide next an alternative result which does not require Assumption 3 and only requires a weaker version of Assumption 4, but also yields weaker conclusions.

Assumption 5 *It holds that*

$$\hat V_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}) - \hat V_{\boldsymbol{\theta}_p}(\mathbf{s}) \le \beta(\Delta_p), \tag{60}$$

for almost all $\mathbf{s}$ *with* β *a* $\mathcal{K}$ *function.*

Before stating the theorem, we observe that Assumption 5 can be seen as a slight relaxation of Assumption 4.

Theorem 6 *Suppose that Assumption 5 holds and assume that each parameter is given by* (26) *or* (38) *such that* $\boldsymbol{\theta}_p \in \boldsymbol{\Theta}_{\mathrm{L}} \cap \boldsymbol{\Theta}_{\mathrm{F}} \cap \boldsymbol{\Theta}_{\mathcal{D}}$. *Then the closed-loop system is ISS.*

PROOF. By assumption, we have that each $\hat V_{\boldsymbol{\theta}_p}$ is upper and lower bounded by $\mathcal{K}_\infty$ functions. Moreover,

$$\hat V_{\boldsymbol{\theta}_p}(\mathbf{s}_+) - \hat V_{\boldsymbol{\theta}_p}(\mathbf{s}) \le -(1-\gamma)\hat V_{\boldsymbol{\theta}_p}(\mathbf{s}) + \delta_{\boldsymbol{\theta}_p},$$

which, using (60), entails that

$$\hat V_{\boldsymbol{\theta}_{p+1}}(\mathbf{s}_+) - \hat V_{\boldsymbol{\theta}_p}(\mathbf{s}) \le -(1-\gamma)\hat V_{\boldsymbol{\theta}_p}(\mathbf{s}) + \delta_{\boldsymbol{\theta}_p} + \beta(\Delta_p),$$

which is the decrease condition for ISS Lyapunov functions [12]. Consequently, the closed-loop system is ISS with respect to Δ and $\delta_{\boldsymbol{\theta}}$. □

The result of Theorem 6 can be understood as follows: while parameter updates might perturb the closed-loop system and temporarily jeopardize asymptotic stability, the destabilizing effect is bounded and disappears as soon as the parameters are not updated anymore. Consequently, as RL converges the possibly destabilizing perturbations decrease in intensity and eventually vanish. The original stability result of [25] is then recovered.

7 Numerical Examples

In this section we provide numerical examples which illustrate the theoretical developments.

7.1 Recursive Feasibility

We first discuss a simple academic example which is constructed, but allows us to discuss the theory in simple terms. Consider the scalar linear system

$$s_+ = As + Ba + w, \qquad A = 1.1, \qquad B = 0.1,$$

with $w \in [\underline{w}, \overline{w}] := [-0.1, 0.1]$. We construct MPC such that it delivers a policy as close as possible to $-Ks + a^{\mathrm{s}}$, where $\boldsymbol{\theta} = \{K, a^{\mathrm{s}}\}$ are parameters to be adjusted by RL and the state and input must satisfy

$$s \le \overline{s} := 0.1, \qquad a \in [-10, 10 - 0.5K].$$

One can verify that the robust MPC formulation

$$\begin{aligned} \min_u \quad & (u - (a^{\mathrm{s}} - Ks))^2 \\ \text{s.t.} \quad & As + Bu + \overline{w} \le \overline{s}, \\ & u \in [-10, 10 - 0.5K], \end{aligned}$$

guarantees that the state constraint $s \le \overline{s}$ is never violated with the given dynamics and process noise. The stage cost $\ell(s,a) = (s-40)^2 + 10^{-4}a^2$ should be minimized by RL, and the MPC region of attraction at convergence must include the interval $[s_{\mathrm{b}}^0, s_{\mathrm{b}}^1] = [0, 0.1]$.

We consider a discount factor $\gamma = 0.9$ and solve the problem by applying constrained policy gradient to the exact total expected cost J. At each policy gradient iteration p we solve the problem

$$\begin{aligned} \boldsymbol{\theta}_{p+1} := \arg\min_{\boldsymbol{\theta}} \; & 0.5\|\boldsymbol{\theta} - \boldsymbol{\theta}_p\|_2^2 + \alpha \nabla_{\boldsymbol{\theta}} J^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_p) \\ \text{s.t.} \; & As_{\mathrm{b}}^j + Ba_{\mathrm{b}}^j + \overline{w} \le \overline{s}, & j = 0, 1, \\ & a_{\mathrm{b}}^j := \left[-Ks_{\mathrm{b}}^j + a^{\mathrm{s}}\right]_{-10}^{10-0.5K}, & j = 0, 1, \\ & A - BK \in [-1+\epsilon, 1-\epsilon], \end{aligned}$$

where $[\cdot]_a^b := \max(a, \min(\cdot, b))$, $\epsilon = 10^{-6}$, $\alpha = 1$, and we computed $\nabla_{\boldsymbol{\theta}} J$ using the deterministic policy gradient theorem to compute the gradient in an actor-critic framework [26].

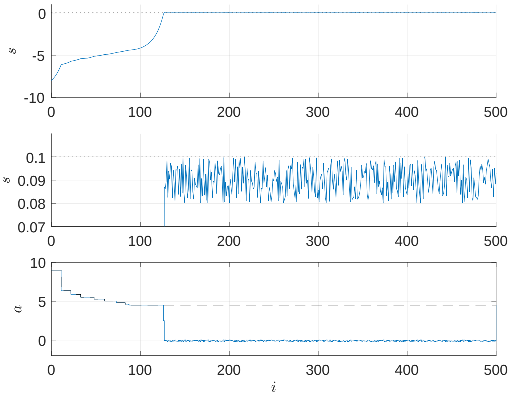

Fig. 1. Closed-loop simulation with parameter updates relying on backtracking. Top plot: state trajectory. Middle plot: state trajectory zoom. Bottom plot: control trajectory (blue line) and upper bound $10 - 0.5K$ (dashed black line).

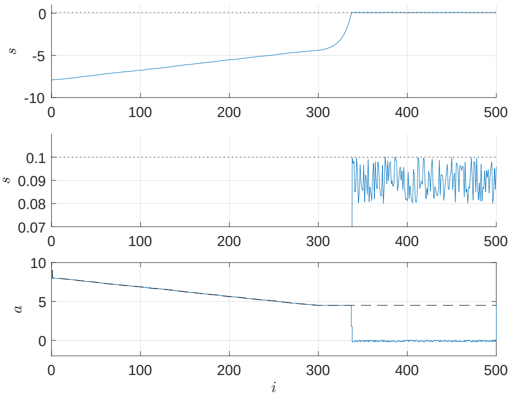

Fig. 2. Closed-loop simulation with trajectory-constrained parameter updates. Top plot: state trajectory. Middle plot: state trajectory zoom. Bottom plot: control trajectory (blue line) and upper bound $10 - 0.5K$ (dashed black line).

We initialize the problem with $K = 2$, $a^{\mathrm{s}} = 0$. The problem converges in one iterate to the optimal solution $K_\star = 11$, $a^{\mathrm{s}} = 0.9$, but, depending on the initial state, the solution cannot be immediately applied to the system. Indeed, the region of attraction for the initial guess is the interval $\mathcal{S}_0 := [-8.9, 0.1]$, while for the optimal solution the region of attraction is the interval $\mathcal{S}_\star := [-4.4, 0.1]$. We display in Figure 1 a simulation starting from $s = -8$ and using a backtracking strategy (Algorithm 1) with $n = 1$, i.e., if the solution is not feasible, α is reduced with $\rho = 0.9$. One can observe that, in the beginning, parameter $\boldsymbol{\theta}$ is not updated until: (a) α becomes smaller and, consequently, the region of attraction becomes larger; and (b) the state approaches the region of attraction.

We also performed a simulation in which the update was done by trajectory-constrained parameter updates, where the following problem was solved

$$\begin{aligned}
\boldsymbol{\theta}_{i+1} := \arg\min_{\boldsymbol{\theta}}\ & 0.5\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\|_2^2 + \alpha \nabla_{\boldsymbol{\theta}} J^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_i) \\
\text{s.t. } & A s_{\mathrm{b}}^j + B a_{\mathrm{b}}^j + \overline{w} \le \overline{s}, && j = 0, 1, \\
& a_{\mathrm{b}}^j := \left[-K_i s_{\mathrm{b}}^j + a_i^{\mathrm{s}} \right]_{-10}^{10-0.5K}, && j = 0, 1, \\
& A - BK \in [-1 + \epsilon, 1 - \epsilon], \\
& s_{\mathrm{wc}}^j \in \mathcal{S}_{\boldsymbol{\theta}}, && j = 0, 1,
\end{aligned}$$

where $\mathcal{S}_{\boldsymbol{\theta}}$ is the region of attraction given $\boldsymbol{\theta}$, and s_{wc}^j are the one-step worst-case state realizations which, in this specific case, are given by

$$\begin{aligned}
s_{\mathrm{wc}}^0 &:= A s_i + B \left[-K_i s_i + a_i^{\mathrm{s}} \right]_{-10}^{10-0.5K_i} + \underline{w}, \\
s_{\mathrm{wc}}^1 &:= A s_i + B \left[-K_i s_i + a_i^{\mathrm{s}} \right]_{-10}^{10-0.5K_i} + \overline{w}.
\end{aligned}$$

The closed-loop trajectories are shown in Figure 2, where one can see that the convergence is slower than with backtracking, since the parameters are updated at each time, which reduces the maximum implementable control and, therefore, makes the convergence to the optimal operating set slower. Nevertheless, also in this case we recover the optimal solution $K_\infty = K_\star$, $a_\infty^{\mathrm{s}} = a_\star^{\mathrm{s}}$.

7.2 Value Function

We consider now the linear system with dynamics and stage cost

$$\begin{aligned}
\mathbf{s}_+ &= \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \mathbf{s} + \begin{bmatrix} 0.05 \\ 0.1 \end{bmatrix} \mathbf{a} + \mathbf{w}, \\
\ell(\mathbf{s}, \mathbf{a}) &= \begin{bmatrix} \mathbf{s} - \mathbf{s}^{\mathrm{r}} \\ \mathbf{a} - \mathbf{a}^{\mathrm{r}} \end{bmatrix}^\top \mathrm{diag}\left(\begin{bmatrix} 1 \\ 0.01 \\ 0.01 \end{bmatrix} \right) \begin{bmatrix} \mathbf{s} - \mathbf{s}^{\mathrm{r}} \\ \mathbf{a} - \mathbf{a}^{\mathrm{r}} \end{bmatrix},
\end{aligned}$$

where $\mathbf{s} = (p, v)$ and $\mathbf{s}^{\mathrm{r}} = (-3, 0)$, $\mathbf{a}^{\mathrm{r}} = 0$. We formulate a problem with prediction horizon $N = 50$ and introduce the state and control constraints $-\mathbf{1} \le \mathbf{s} \le \mathbf{1}$, $-10 \le \mathbf{a} \le 10$. The real noise set is selected as a regular octagon, and we parametrize $\mathbb{W}_{\boldsymbol{\omega}}$ as a polytope with 4 facets.

We formulate tube based MPC as

$$
\begin{aligned}
&Q_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) := \\
&\quad \min_{\mathbf{z}} \ \sum_{k=0}^{N-1} \left\| \begin{matrix} \mathbf{x}_k - \mathbf{x}_\mathrm{r} \\ \mathbf{u}_k - \mathbf{u}_\mathrm{r} \end{matrix} \right\|_H^2 + \left\| \mathbf{x}_N - \mathbf{x}_\mathrm{r} \right\|_P^2 \\
&\qquad\qquad + \left\| \mathbf{x}_0 \right\|_\Lambda^2 + \boldsymbol{\lambda}^\top \mathbf{x}_0 + l && \text{(61a)} \\
&\quad \text{s.t.} \ \ \mathbf{x}_0 = \mathbf{s}, \qquad \mathbf{u}_0 = \mathbf{a}, && \text{(61b)} \\
&\qquad \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{b}, \quad k \in \mathbb{I}_0^{N-1}, && \text{(61c)} \\
&\qquad C\mathbf{x}_k + D\mathbf{u}_k + \mathbf{c}_k \leq \mathbf{0}, \qquad k \in \mathbb{I}_0^{N-1}, && \text{(61d)} \\
&\qquad G\mathbf{x}_N + \mathbf{g} \leq \mathbf{0}, && \text{(61e)}
\end{aligned}
$$

where one must enforce that the system dynamics (61c) and a parametrized compact uncertainty set $\mathbb{W}_{\boldsymbol{\omega}}$ are such that $\mathbf{s}_+ - (A\mathbf{s}+B\mathbf{a}+\mathbf{b}) \in \mathbb{W}_{\boldsymbol{\omega}}$. This issue has been discussed in detail in [31], where the set is parametrized as the polyhedron $\mathbb{W}_{\boldsymbol{\omega}} := \{ \ \mathbf{w} \mid M\mathbf{w} \leq \mathbf{m} \ \}$ and the following set membership constraint is imposed on $\boldsymbol{\omega} = (M, \mathbf{m})$ for all past samples $\mathbf{s}_{i+1}, \mathbf{s}_i, \mathbf{a}_i$, $i \in \mathcal{I}$:

$$M(\mathbf{s}_{i+1} - (A\mathbf{s}_i + B\mathbf{a}_i + \mathbf{b})) \leq \mathbf{m}, \qquad \forall \ i \in \mathcal{I}.$$

Then, $\mathbf{c}_k$ is computed by tightening the original constraints $C\mathbf{s} + D\mathbf{a} + \hat{\mathbf{c}} \leq 0$, so as to guarantee that, for any process noise $\mathbf{w} \in \mathbb{W}_{\boldsymbol{\omega}}$, the constraints are satisfied. Moreover, parameters $\mathbf{x}_\mathrm{r}, \mathbf{u}_\mathrm{r}$ must be a steady-state for the system dynamics (61c), i.e.,

$$(A - I)\mathbf{x}_\mathrm{r} + B\mathbf{u}_\mathrm{r} = \mathbf{0}.$$

Finally, G and $\mathbf{g}$ must be selected such that they define a robust positively invariant terminal set for the feedback law $\mathbf{u} = -K(\mathbf{x} - \mathbf{x}_\mathrm{r}) + \mathbf{u}_\mathrm{r}$, with K the solution to the LQR formulated with A, B, H, P. The vector of MPC parameters is then defined as

$$\boldsymbol{\theta} = \{\Lambda, \lambda, l, H, \mathbf{x}_\mathrm{r}, \mathbf{u}_\mathrm{r}, M\}, \qquad (62)$$

and we consider K, P, $\mathbf{c}_k$, G, $\mathbf{g}$ as functions of these parameters. Vector $\mathbf{m}$ can also be included in $\boldsymbol{\theta}$, but, as discussed in [31] this is not necessary. Matrices C, D and vector $\bar{\mathbf{c}}$ are assumed to be known. Finally, A, B, $\mathbf{b}$ could in principle also be included in the parameter vector $\boldsymbol{\theta}$. However, as discussed in [31] this makes the safe RL problem much harder to formulate and solve, since it obliges one to store very large amounts of data and formulate an equally large amount of constraints.

The set of parameters guaranteeing safety and stability then becomes

$$
\begin{aligned}
\Theta := \{ \ \boldsymbol{\theta} \mid \ & H \succ 0, \\
& M(\mathbf{s}_{i+1} - (A\mathbf{s}_i + B\mathbf{a}_i + \mathbf{b})) \leq \mathbf{m}, \ \forall \ i \in \mathcal{I}, \\
& (A - I)\mathbf{x}_\mathrm{r} + B\mathbf{u}_\mathrm{r} = \mathbf{0}, \\
& \exists \ \mathbf{x} \text{ s.t. } G\mathbf{x} \leq \mathbf{g} \ \},
\end{aligned}
$$

i.e., the noise set must include all observed noise samples, the reference must be a steady-state of the system and the terminal set must be nonempty. This last condition also entails that the MPC domain is nonempty.

We update $\boldsymbol{\theta}$ using a batch Q learning approach with batches of horizon $N_\mathrm{b} = 20$ with learning rate $\alpha = 0.1$, using the backtracking strategy.

We simulated the system starting from state $\mathbf{s}_0 = (0.8, 0)$. The backtracking strategy never rejected nor reduced any step. The resulting closed-loop trajectory is displayed in Figure 3, together with the reference, maximum robust positive invariant (MRPI) and terminal sets at the beginning and end of the simulation, as well as the minimum robust positive invariant (mRPI) sets throughtout the simulation. We display the noise set approximation at the end of the simulation in Figure 4, and the evolution throughout the RL epochs of the parameter $\boldsymbol{\theta}$ and the average TD error in each batch in Figure 5. We display the MPC Lyapunov functions $\hat{V}_{\boldsymbol{\theta}}$ and W in time in Figure 6. One can see that in the beginning $\hat{V}_{\boldsymbol{\theta}}$ sometimes increases upon parameter updates, but decreases inside each batch. Note that this result is perfectly in line with Theorem 5 and Remark 3. After the displayed time interval, the Lyapunov function $\hat{V}_{\boldsymbol{\theta}}$ was always 0, i.e., the state trajectory remained inside the mRPI set, even when this set was updated by a parameter change. Some words are due in order to discuss function W: as pointed out in Remark 4, in practice one can at best expect that the parameters converge to a neighborhood of the optimal ones. Therefore, we selected $\boldsymbol{\theta}_\star$ as the average of $\boldsymbol{\theta}$ over the last 100 epochs, when, as shown in Figure 5, the parameter is at convergence. We observed that $\zeta = 0.1$ was sufficiently high to satisfy the conditions of Theorem 5. As one can see in Figure 6, differently from $\hat{V}_{\boldsymbol{\theta}}$, the obtained W is decreasing also in the first epochs. Finally, we show the performance J in terms of total discounted cost over each batch in Figure 7. One can see that after a short transient the performance reaches convergence and does not improve anymore.

8 Conclusions

This paper discusses how to implement Learning-based adaptations of a robust MPC scheme in order to improve its closed-loop performance while maintaining the stability and safety of the control policy. We show in particular that these requirements can be treated via constraints on the learning steps, and parameter update conditions that are fairly simple to verify, and that can be implemented online in real time. We additionally establish that the proposed approach ensures that the update conditions are not blocking the learning process, in the sense that they are met in final time with probability one. We finally show that under some conditions on

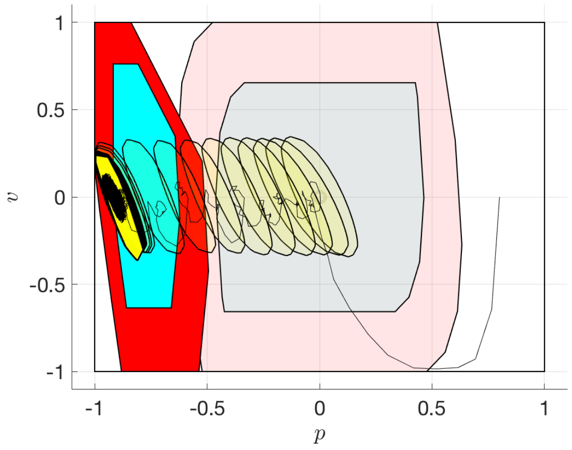

Fig. 3. MRPI (red), terminal (cyan) sets and reference $\mathbf{x}^{\mathrm{r}}$ (black and grey circle) at the beginning and end of the learning process; state trajectoyr (black line) and mRPI sets (yellow) at each time instant.

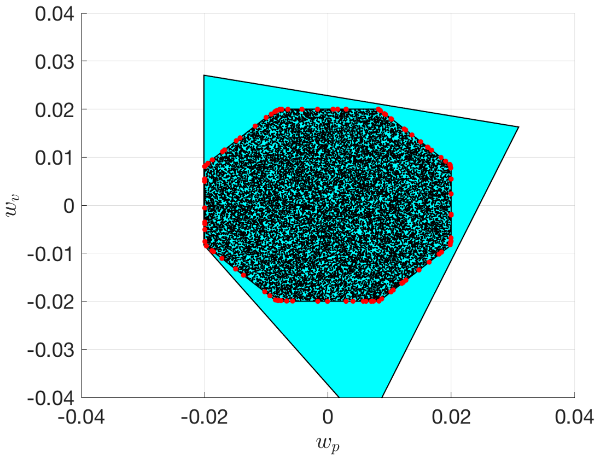

Fig. 4. True process noise set (transparent octogon), noise samples (black dots), their convex hull (red dots) and noise set parametrized by matrix M (cyan).

the learning process, a form of stability of the resulting learning-based robust MPC scheme is guaranteed in the state-parameter space. The proposed approaches are illustrated in two simulated examples.

References

[1] S. Abdufattokhov, M. Zanon, and A. Bemporad. Learning Convex Terminal Costs for Complexity Reduction in MPC. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2163–2168. 2021.

[2] 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 NIPS*, NIPS'18, pages 8299–8310, USA, 2018. Curran Associates Inc.

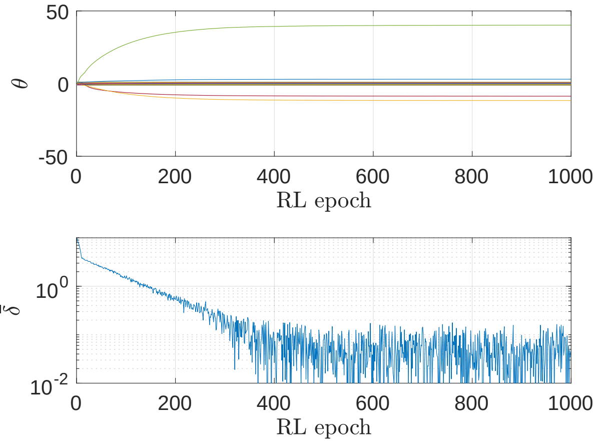

Fig. 5. Top plot: parameter evolution through the epochs. Bottom plot: TD error through the epochs.

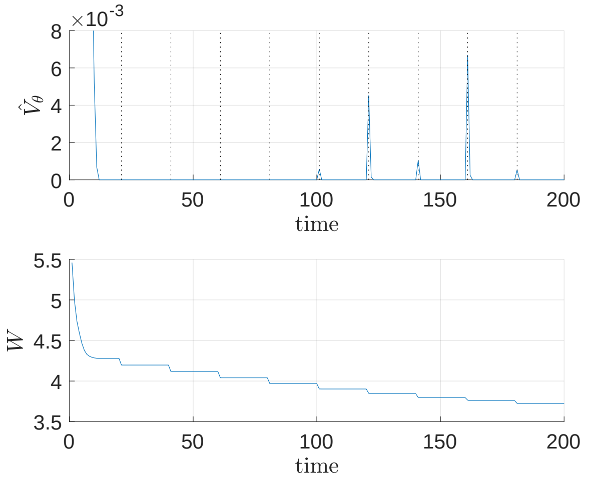

Fig. 6. Top figure: Lyapunov function $\hat{V}_{\boldsymbol{\theta}}$ over the first epochs. Bottom plot: Lyapunov function W over time.

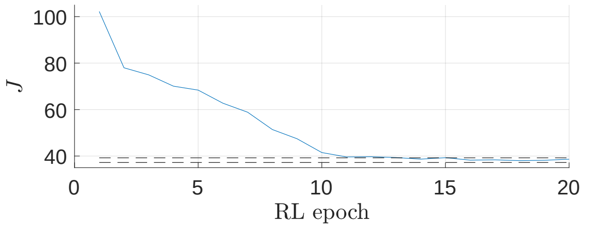

Fig. 7. Performance J in terms of total discounted cost for each batch. The dashed lines indicate the maximum and minimum of J over all future times.

[3] A. Aswani, H.o Gonzalez, S. S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216 – 1226, 2013.

[4] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe Model-based Reinforcement Learning with Stability Guarantees. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

editors, *Advances in Neural Information Processing Systems 30*, pages 908–918. Curran Associates, Inc., 2017.

[5] D.P. Bertsekas and I.B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Transactions on Automatic Control*, 16:117–128, 1971.

[6] L. Chisci, J.A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37:1019–1028, 2001.

[7] A.V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press, New York, 1983.

[8] Chris Gaskett. Reinforcement learning under circumstances beyond its control. In *International conference on computational intelligence, robotics and autonomous systems*, 2003.

[9] S. Gros and M. Zanon. Data-Driven Economic NMPC Using Reinforcement Learning. *IEEE Transactions on Automatic Control*, 65(2):636–648, Feb 2020.

[10] Matthias Heger. Consideration of Risk in Reinforcement Learning. In *Machine Learning Proceedings 1994*, pages 105–111. Elsevier, 1994.

[11] Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-Based Model Predictive Control: Toward Safe Learning in Control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[12] Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[13] Johannes Köhler, Peter Kötting, Raffaele Soloperto, Frank Allgöwer, and Matthias A. Müller. A Robust Adaptive Model Predictive Control Framework for Nonlinear Uncertain Systems. *International Journal of Robust and Nonlinear Control*, 31(18):8725–8749, 2021.

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

[15] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3):32–50, 2009.

[16] 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*, 32(6):76–105, 2012.

[17] D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. MPC for Tracking Piecewise Constant References for Constrained Linear Systems. *Automatica*, 44(9):2382–2387, 2008.

[18] D. Masti, M. Zanon, and A. Bemporad. Tuning LQR Controllers: a Sensitivity-Based Approach. *IEEE Control Systems Letters*, 6:932–937, 2022. Also in 60th IEEE Conf. on Decision and Control, Austin, TX, December 13-15, 2021.

[19] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

[20] D.Q. Mayne, M.M. Seron, and S.V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41:219–224, 2005.

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

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

[23] 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*, 35(13):1547–1563, 2016.

[24] Iason Papaioannou, Wolfgang Betz, Kilian Zwirglmaier, and Daniel Straub. MCMC algorithms for Subset Simulation. *Probabilistic Engineering Mechanics*, 41:89–103, 2015.

[25] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, 2nd edition, 2017.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of ICML*, ICML'14, pages I–387–I–395, 2014.

[27] R. Soloperto, M. Müller, and F. Allgöwer. Guaranteed Closed-Loop Learning in Model Predictive Control.

[28] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*, pages 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

[29] Mario E. Villanueva, Rien Quirynen, Moritz Diehl, Benoît Chachuat, and Boris Houska. Robust MPC via Min-Max Differential Inequalities. *Automatica*, 77:311–321, 2017.

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

[31] M. Zanon and Gros. Safe Reinforcement Learning Using Robust MPC. *Transaction on Automatic Control*, 66(8):3638–3652, 2021.

[32] M. Zanon and S. Gros. On the Similarity Between Two Popular Tube MPC Formulations. In *Proceedings of the European Control Conference*, pages 651–656, 2021.

[33] M. Zanon, S. Gros, and A. Bemporad. Practical Reinforcement Learning of Stabilizing Economic MPC. In *Proceedings of the European Control Conference*, pages 2258–2263, 2019.

[34] Mengjia Zhu, Dario Piga, and Alberto Bemporad. C-GLISp: Preference-Based Global Optimization Under Unknown Constraints With Applications to Controller Calibration. *IEEE Transactions on Control Systems Technology*, 2022. (in press).

[35] Konstantin M. Zuev. *Subset Simulation Method for Rare Event Estimation: An Introduction*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2021.