

# Robust Learning Model Predictive Control for Linear Systems Performing Iterative Tasks

Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli

**Abstract**—A robust Learning Model Predictive Controller (LMPC) for uncertain systems performing iterative tasks is presented. At each iteration of the control task the closed-loop state, input, and cost are stored and used in the controller design. This paper first illustrates how to construct robust control invariant sets and safe control policies exploiting historical data. Then, we propose an iterative LMPC design procedure, where data generated by a robust controller at iteration  $j$  are used to design a robust LMPC at the next iteration  $j + 1$ . We show that this procedure allows us to iteratively enlarge the domain of the control policy and it guarantees recursive constraints satisfaction, input to state stability, and performance bounds for the certainty equivalent closed-loop system. The use of different feedback policies along the horizon is the key element of the proposed design. The effectiveness of the proposed control scheme is illustrated on a linear system subject to bounded additive disturbances.

## I. INTRODUCTION

Exploiting historical data to iteratively improve the performance of Model Predictive Controllers (MPC) has been an active theme of research over the past few decades [1]–[25]. The key idea is to use the stored state, input, and cost data to compute at least one of the following control design elements: *i)* a *model* to predict the system trajectory for a given initial state and input sequence, *ii)* a *safe set* of states from which the control task can be completed using a known safe policy and *iii)* a *value function*, which for a given safe policy, maps each state of the safe set to the closed-loop cost of completing the task.

Policy evaluation strategies used to estimate value functions from historical data are studied in Approximate Dynamic Programming (ADP) and Reinforcement Learning (RL) [26]–[28]. For instance, direct strategies compute the estimate value function which best fits the realized closed-loop cost over the stored states. On the other hand, in indirect strategies the estimate value function is computed by iteratively minimizing the temporal difference [29], [30]. A survey on policy evaluation strategies goes beyond the scope of this paper, we refer the reader to [26], [27] for a comprehensive review on this topic.

The integration of MPC with system identification strategies used to estimate the prediction model has been extensively studied in literature [3]–[14]. In adaptive MPC strategies [18]–[25], set-membership approaches are used to identify the set of possible parameters and/or the domain of the uncertainty which characterize the system’s model. Afterwards, robust MPC strategies for additive [31] or parametric [32], [33] uncertainty are used to guarantee recursive constraint satisfaction.

Another strategy to identify the system dynamics is to fit a Gaussian Process (GP) to experimental data [4]–[8]. GP can be used to identify a nominal model and confidence bounds, which may be used to tighten the constraint set over the planning horizon. This strategy provides high-probability safety guarantees [5], [8]. The effectiveness of GP-based strategies on experimental platforms has been shown in [6], [7], where an MPC is used to race a 1/43-scale vehicle and to safely fly a drone. Regression strategies may also be used to identify the system model [9], [14]. For instance, the authors in [9] used a linear regression strategy to identify both a nominal model and the disturbance domain used for robust MPC design. In [14], we used local linear regression to identify the system model used by the controller, which was able to drive a 1/10-scale race car at the limit of handling.

Model-based and data-based approaches for computing safe sets have also been proposed in literature [10], [13], [34]–[41]. In reachability-based strategies safe sets are computed solving a two players game between the controller and the disturbance [34]–[36]. Furthermore, these strategies provide a control policy, which can be used to guarantee safety by robustly constraining the evolution of the system within the safe set [34]. Also viability theory may be used to compute safe sets [37]. The authors in [37] showed how to compute an inner approximation of the viability kernel and demonstrated the effectiveness on a RC-car set-up. In [38] the authors showed how to compute safe sets for uncertain systems exploiting data from a robust controller, afterwards they used the safe set in a linear model predictive safety certification framework. In [10], [13] the authors computed safe sets and the associated control policy combining stored trajectories with polyhedral and ellipsoidal invariant sets. Another approach is proposed in [39] where the stored trajectories are mirrored to construct invariant sets. Finally, in [40], [41] we have shown how data from a deterministic system can be trivially used to compute safe sets. However, these strategies cannot be used to compute safe sets for uncertain systems.

In this paper, we present an iterative *robust* Learning Model Predictive Control (LMPC) design procedure for uncertain systems. We consider control tasks where the goal is to minimize a given cost function while satisfying state and input constraints. An iteration is a finite time execution of a control task from a given initial condition. We assume that at each iteration the constraint set and cost function are unchanged, but the initial condition and duration can be selected by the control designer. At each iteration we exploit historical data and the LMPC policy from previous iterations to construct robust safe sets and approximations to the value function. Furthermore, we show how to compute a safe control policy which is defined over the robust safe set and it is used in the

U. Rosolia, X. Zhang and F. Borrelli are with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94701, USA {ugo.rosolia, xiaojing.zhang, fborrelli}@berkeley.edu

LMPC design. In particular, the controller plans the system trajectory using either a safe policy or a disturbance feedback policy. We show that the proposed strategy guarantees that: i) state and input constraints are recursively satisfied, ii) the closed-loop system is Input-to-State Stable (ISS) and iii) the performance of the certainty equivalent system is bounded by a function  $Q^j(\cdot)$  which is non-increasing with the iteration index (i.e.  $Q^{j+1}(\cdot) \leq Q^j(\cdot)$ ). Finally, we show that the proposed iterative design procedure can be used to collect data on progressively larger regions of the state space. Thus, in applications where a conservative robust controller is available, our iterative algorithm may be used to improve the closed-loop performance and to enlarge the region of attraction.

Compared with standard robust MPC strategies [31], [42]–[45], where the terminal cost and constraint set used in the design are fixed, the proposed strategy iteratively updates these components used to synthesize the control policy. This strategy, which is tailored to iterative tasks, allows us to iteratively enlarge the region of attraction of the controller and to improve the performance of the certainty equivalent closed-loop system. Compared to the robust certainty equivalent LMPC in [46], where we constructed a control invariant set and a control Lyapunov function for the certainty equivalent system, in this work we compute a *robust* control invariant set and a control Lyapunov function for the uncertain system. We underline that extending the nominal and certainty equivalent LMPC approaches [40], [41], [46] to uncertain systems is not straightforward. In fact, as it will be clear later in this paper, standard shifting MPC arguments for proving recursive robust constraint satisfaction do not apply. In particular, the use of different feedback policies along the prediction horizon is one of the key elements of the proposed control design which is necessary for providing robustness guarantees.

This paper is organized as follows: in Section II we recall some definitions from set theory and the definition of Input-to-State Stability (ISS). Section III describes the challenges of learning safe sets from stored data of uncertain systems. Section IV describes the problem formulation and design requirements. The strategy proposed to compute safe sets and the  $Q$ -function is described in Section V. Section VI describes the control design strategy and Section VII illustrates the controller properties. A strategy to approximate the region of attraction of the controller is presented in Section VIII. Finally, in Section IX we show the effectiveness of the proposed controller on a double integrator system subject to bounded additive uncertainty.

## II. TECHNICAL BACKGROUND

In this section we recall some definitions from set theory [47, Chapter 10], which will be used later in this work.

**Definition 1 (Positive Invariant Set):** A set  $\mathcal{O} \subseteq \mathcal{X}$  is said to be a positive invariant set for the autonomous system  $x_{t+1} = Ax_t$ , if

$$x \in \mathcal{O} \rightarrow Ax \in \mathcal{O}.$$

**Definition 2 (Robust Positive Invariant Set):** A set  $\mathcal{O} \subseteq \mathcal{X}$  is said to be a robust positive invariant set for the uncertain

autonomous system  $x_{t+1} = Ax_t + w_t$ , with  $w_t \in \mathcal{W}$  if

$$x \in \mathcal{O} \rightarrow Ax + w \in \mathcal{O}, \forall w \in \mathcal{W}.$$

**Definition 3 (Robust Control Positive Invariant Set):** A set  $\mathcal{C} \subseteq \mathcal{X}$  is said to be a robust control positive invariant set for the uncertain system  $x_{t+1} = Ax_t + Bu_t + w_t$ , with  $w_t \in \mathcal{W}$  and  $u_t \in \mathcal{U}$ , if

$$x \in \mathcal{C} \rightarrow \exists u \in \mathcal{U} : Ax + Bu + w \in \mathcal{C}, \forall w \in \mathcal{W}.$$

**Definition 4 (Robust Successor Set):** Given a control policy  $\pi(\cdot)$  and the closed-loop system  $x_{t+1} = Ax_t + B\pi(x_t) + w_t$ , we denote the robust successor set from the set  $\mathcal{S}$  as

$$\text{Succ}(\mathcal{S}, \mathcal{W}, \pi) = \{x_{t+1} \in \mathbb{R}^n : \exists x_t \in \mathcal{S}, \exists w_t \in \mathcal{W} \text{ such that } x_{t+1} = Ax_t + B\pi(x_t) + w_t\}.$$

Given the initial state  $x_t$ , the robust successor set  $\text{Succ}(x_t, \mathcal{W}, \pi)$  collects the states that the uncertain autonomous system may reach in one time step.

**Definition 5 (N-Step Robust Reachable Set):** Given a control policy  $\pi(\cdot)$  and the closed-loop system  $x_{t+1} = Ax_t + B\pi(x_t) + w_t$  with  $w_t \in \mathcal{W}$  for all  $t \geq 0$ , for  $k = \{0, \dots, N-1\}$  we recursively define the  $N$ -step robust reachable set from the set  $\mathcal{S}$  as

$$\mathcal{R}_{t \rightarrow t+k+1}(\mathcal{S}, \pi) = \text{Succ}(\mathcal{R}_{t \rightarrow t+k}(\mathcal{S}, \pi), \mathcal{W}, \pi)$$

where  $\mathcal{R}_{t \rightarrow t}(\mathcal{S}, \pi) = \mathcal{S}$ . Robust reachable sets are also referred to as forwards reachable sets.

Given a linear time-invariant system, the  $N$ -Step robust reachable set  $\mathcal{R}_{t \rightarrow t+N}(\mathcal{S}, \pi)$  collects the states which can be reached from the set  $\mathcal{S}$  in  $N$ -steps. Finally, we recall the definition of Input-to-State Stability (ISS) of a robust invariant set [48], which extends the more widely known notion of ISS of an equilibrium point [49]–[52]. We use the standard function classes  $\mathcal{K}$ ,  $\mathcal{K}_\infty$ , and  $\mathcal{KL}$  notation (see [53]) and we define the distance from a point  $x \in \mathbb{R}^n$  to a set  $\mathcal{O} \subseteq \mathbb{R}^n$  as

$$|x|_{\mathcal{O}} \triangleq \inf_{d \in \mathcal{O}} \|x - d\|_2.$$

**Definition 6 (Input to State Stability (ISS) [48]):** Let  $\mathcal{O}$  be a robust positive invariant set for the autonomous system  $x_{t+1} = Ax_t + B\pi(x_t) + w_t$  with  $w_t \in \mathcal{W}$ . We say that the closed-loop system is ISS with respect to  $\mathcal{O}$  if for all  $w_t \in \mathcal{W}$ ,  $t \geq 0$ , and  $x_0 \in \mathcal{X}$  we have:

$$|x_{t+1}|_{\mathcal{O}} \leq \beta(|x_0|_{\mathcal{O}}, t+1) + \gamma(\sup_{k \in \{0, \dots, t\}} \|w_k\|),$$

where  $\beta(\cdot, \cdot)$  is a class- $\mathcal{KL}$  function and  $\gamma(\cdot)$  is a class- $\mathcal{K}$  function.

## III. COMPUTING INVARIANT SETS FROM DATA

In order to motivate our work, this section highlights the challenges associated with the computation of invariant sets from data. First, we recall from [40], [41] how historical data can be used to compute invariant sets for deterministic systems. Consider the discrete time linear system

$$\bar{x}_{t+1}^j = A\bar{x}_t^j + B\pi^j(\bar{x}_t^j)$$

where  $\pi^j(\cdot)$  is a feedback policy known only along the  $j$ th stored trajectory  $\bar{\mathbf{x}}^j = [\bar{x}_0^j, \dots, \bar{x}_t^j, \dots, \bar{x}_{T_j}^j]$ . Assume that

$\pi^j(\cdot)$  is able to execute the desired task safely, meaning that  $\bar{x}_{T^j} \in \mathcal{O}$ . At any iteration  $i > j$  and time  $k \geq 0$ , if the system state  $x_k^i$  equals a state  $x_t^j$  which has been visited at the previous  $j$ th iteration, then the feedback policy  $\pi^j(\cdot)$  will drive the system along the  $j$ th trajectory. This obvious fact is a consequence of the system being deterministic. More importantly, as the policy  $\pi^j(\cdot)$  brings the system to the invariant set  $\mathcal{O}$ , the convex hull of visited states and  $\mathcal{O}$  is a control invariant set. Therefore, invariant sets for deterministic systems can be easily built from data.

In contrast, when dealing with *uncertain systems*, the set of visited states is not an invariant set. In fact, consider the discrete time uncertain system

$$x_{t+1}^j = Ax_t^j + B\pi^j(x_t^j) + w_t^j,$$

where the random disturbance  $w_t^j$  belongs to the set  $\mathcal{W}$  and the  $j$ th stored trajectory is  $\mathbf{x}^j = [x_0^j, \dots, x_t^j, \dots, x_{T^j}^j]$ . Assume that  $\pi^j(\cdot)$  is able to execute the desired task safely at iteration  $j$ . Notice that the stored trajectory  $\mathbf{x}^j$  is associated with a specific disturbance realization  $[w_0^j, \dots, w_t^j, \dots]$ . For this reason at any iteration  $i > j$  and time  $k \geq 0$ , if the system state  $x_k^i$  equals a state  $x_t^j$  that has been visited, applying the feedback policy  $\pi^j(\cdot)$  may drive the system to a state neither stored nor safe, due to a potentially different disturbance realization  $[w_0^i, \dots, w_t^i, \dots]$ . In conclusion, the set of visited states cannot be naively exploited to compute invariant sets.

Furthermore, we underline that even if a control invariant set for uncertain systems can be computed from data, its use for MPC design is not straightforward. This issue will become clear later in this paper. In the following, we first present a strategy to construct robust control invariant sets using stored data from a linear uncertain system. Afterwards, we leverage these sets to iteratively synthesize robust LMPC policies.

#### IV. PROBLEM FORMULATION

We consider a discrete time uncertain system of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad (1)$$

where  $x_t \in \mathbb{R}^n$  and  $u_t \in \mathbb{R}^d$  are the state and the input at time  $t$  and the matrices  $A$  and  $B$  are known. The disturbances  $w_t^j$  are zero mean independent and identically distributed (*i.i.d.*) with bounded support  $\mathcal{W} \subset \mathbb{R}^n$ .

*Assumption 1:* The disturbance's support  $\mathcal{W}$  is a compact polytope described by  $l$  vertices  $\{v_w^1, \dots, v_w^l\}$  and it contains the origin.

Furthermore, the system is subject to the following convex constraints on states and inputs:

$$x_t \in \mathcal{X} \text{ and } u_t \in \mathcal{U}, \quad \forall t \geq 0, \quad (2)$$

where the sets  $\mathcal{X}$  and  $\mathcal{U}$  contain the origin and are assumed to be compact.

In this paper, we consider control tasks where we would like to steer system (1) towards the set  $\mathcal{O}$ , while minimizing the summation of the stage cost  $h: \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$  and satisfying constraints (2). Throughout the paper we make the following assumptions.

*Assumption 2:* The set  $\mathcal{O} \subset \mathcal{X} \subset \mathbb{R}^n$  is a robust positive invariant set for the autonomous system  $x_{t+1} = (A+BK)x_t + w_t$  with  $w_t \in \mathcal{W}$ . Furthermore,  $\mathcal{O}$  is a polyhedron defined through its vertices  $\{v_o^1, \dots, v_o^m\}$  and

$$K\mathcal{O} = \{u \in \mathbb{R}^d : \exists x \in \mathcal{O}, u = Kx\}.$$

*Assumption 3:* The stage cost  $h(\cdot, \cdot)$  is continuous and jointly convex in its arguments. Furthermore, we assume that for all  $x \in \mathbb{R}^n$ , and for all  $u \in \mathbb{R}^d$  the stage cost satisfies:

$$\alpha_x^l(|x|_{\mathcal{O}}) \leq h(x, 0) \leq \alpha_x^u(|x|_{\mathcal{O}})$$

$$\text{and } \alpha_u^l(|u|_{K\mathcal{O}}) \leq h(0, u) \leq \alpha_u^u(|u|_{K\mathcal{O}}),$$

where  $\alpha_x^u, \alpha_x^l, \alpha_u^u$ , and  $\alpha_u^l \in \mathcal{K}_{\infty}$ .

*Remark 1:* In Assumption 2 a robust invariant  $\mathcal{O}$  is required. In the proposed approach  $\mathcal{O}$  can be a very small neighborhood of the origin. In fact, the iterative nature of the control design will allow us enlarge the closed-loop region of attraction at each iteration.

#### A. Iteration and Control Design Objectives

Iteration  $j$  refers to a finite time execution of the control task from the initial conditions  $x_0^j$ . At each iteration, the initial condition may be different, however the system model (1), the constraints (2), the set  $\mathcal{O}$ , and the cost function  $h(\cdot, \cdot)$  are identical. We denote  $x_t^j$  and  $u_t^j$  as the state and input of the system at time  $t$  of iteration  $j$ , i.e.,

$$x_{t+1}^j = Ax_t^j + Bu_t^j + w_t^j. \quad (3)$$

Furthermore, we define  $T^j$  as the finite time duration of iteration  $j$ . As we will discuss in Section V, the time  $T^j$  is selected by the control designer.

At each iteration  $j$ , our objective is to design a state feedback policy for the uncertain system (3)

$$\pi^j(\cdot) : \mathcal{C}^j \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^d, \quad (4)$$

and the associated region of attraction  $\mathcal{C}^j$  such that at each  $j$ th iteration and for all  $x_0^j \in \mathcal{C}^j \subseteq \mathbb{R}^n$  we have that:

- 1) The certainty equivalent system

$$\bar{x}_{t+1}^j = A\bar{x}_t^j + B\bar{u}_t^j \quad (5)$$

with  $\bar{u}_t^j = \pi^j(\bar{x}_t^j)$  converges asymptotically to the set  $\mathcal{O}$ , i.e.  $\lim_{t \rightarrow \infty} \bar{x}_t^j \in \mathcal{O}$ .

- 2) The closed-loop system  $x_{t+1}^j = Ax_t^j + B\pi^j(x_t^j) + w_t^j$  is Input-to-State Stable (ISS) with respect to the set  $\mathcal{O}$  (see Section II for the definition of ISS).
- 3) The closed-loop state and input constraints are robustly satisfied, namely

$$x_t^j \in \mathcal{X} \text{ and } \pi^j(x_t^j) \in \mathcal{U}, \quad \forall w_t^j \in \mathcal{W}, \quad \forall t \geq 0.$$

- 4) The domain  $\mathcal{C}^j$  of policy  $\pi^j(\cdot)$  does not shrink with the iteration index, i.e.,  $\mathcal{C}^j \subseteq \mathcal{C}^{j+1}$ .
- 5) The iteration cost of the certainty equivalent system (5), defined as

$$J_{0 \rightarrow T^j}^j(\bar{x}_0^j) = \sum_{k=0}^{T^j} h(\bar{x}_k^j, \pi^j(\bar{x}_k^j)),$$

is an upper-bound to the function  $Q^{j-1}(\cdot)$  (i.e.  $J_{0 \rightarrow T^j}^j(x_0^j) \leq Q^{j-1}(x_0^j)$ ), which is non-increasing at each iteration

$$Q^j(\bar{x}) \leq Q^k(\bar{x}), \forall j \geq k.$$

Property 5) implies that, as more data is collected, the upper-bound on the performance of the certainty equivalent closed-loop system is non-increasing.

*Remark 2:* Note that convergence and stability properties 1) and 2) hold for  $t \rightarrow \infty$ , although we study iterations with finite time duration. In particular, an iteration is terminated after  $T^j$  time steps and, therefore, at the end of the  $j$ th iteration the closed-loop system may not reach the set  $\mathcal{O}$ . As we will discuss later on, the duration of an iteration  $T^j$  may affect the closed-loop performance of the controller.

## V. LMPC PRELIMINARIES

In this section, we assume that a robust policy at iteration  $j$  is given and we describe how to compute the terminal cost and constraints used to synthesize the control policy at the next iteration  $j+1$ . In particular, we show how historical data and the  $j$ th robust MPC policy can be used to build a *robust safe set* of states from which the iteration  $j+1$  can be executed.

The iterative synthesis procedure used to update the control policy is described in Section VI. This iterative procedure is based on the robust safe set,  $Q$ -function, and safe policy, which are defined in this section. Their initialization at iteration  $j=0$  is also discussed.

### A. Robust Safe Set

This section shows how to iteratively construct robust control invariant sets. In particular, we run the closed-loop system at iteration  $j$  and we exploit the closed-loop trajectory to construct a robust safe set.

We initialize the *robust convex safe set* at iteration  $j=0$  as

$$\mathcal{CS}^0 = \mathcal{O}. \quad (6)$$

Afterwards, we design the robust  $N$ -steps policy

$$\pi_t^{1,*}(\cdot) = [\pi_{t|t}^{1,*}(\cdot), \dots, \pi_{t+N-1|t}^{1,*}(\cdot)], \quad (7)$$

which steers the following closed-loop system

$$x_{t+1}^1 = Ax_t^1 + B\pi_{t|t}^{1,*}(x_t^1) + w_t \quad (8)$$

to the robust convex safe set  $\mathcal{CS}^0$ . Next, we show that the control policy (7) and the convex safe set (6) can be used to construct the convex safe set at iteration 1.

*Assumption 4:* For all  $t \in \{0, \dots, T^1\}$ , the  $N$ -steps policy  $\pi_t^{1,*}(\cdot)$  in (7) robustly steers the predicted closed-loop system

$$x_{k+1|t}^1 = Ax_{k|t}^1 + B\pi_{k|t}^{1,*}(x_{k|t}^1) + w_{k|t}, \quad \forall k = t, \dots, t+N-1$$

from the state  $x_t^1 \in \mathcal{C}^1$  to the robust convex safe set  $\mathcal{CS}^0 = \mathcal{O}$  in  $N$ -steps, while robustly satisfying state and input constraints (2).

*Remark 3:* The control policy  $\pi_t^{1,*}(\cdot)$ , which satisfies Assumption 4, can be computed using the iterative procedure that we will be describing in Section VI.

*Remark 4:* We underline that  $T^j$  represents the duration of the  $j$ th closed-loop simulation. On the other hand,  $N$  is the length of the prediction horizon associated with the control policy  $\pi_t^{1,*}(\cdot)$ . Notice that  $T^j$  and  $N$  are not related but in general  $N < T^j$ . Indeed, the horizon length  $N$  is usually chosen much smaller than the task duration  $T^j$  to reduce the computational burden associated with the control policy. Finally, we underline that the duration of the control task  $T^j$  is chosen by the control designer.

Let the vectors

$$\begin{bmatrix} x_0^1, \dots, x_{T^1}^1 \\ u_0^1, \dots, u_{T^1}^1 \end{bmatrix} \quad (9)$$

collect states and inputs associated with a simulation of the closed-loop system (8). As we will discuss later on, by linearity of the system, any state in the convex-hull of the closed-loop trajectory in (9) can be robustly steered to  $\mathcal{O}$ . However, the convex hull of the stored states in (9) and  $\mathcal{O}$ , denoted as  $\text{Conv}\{\{\cup_{t=0}^{T^j} x_t^1\} \cup \mathcal{O}\}$ , is not invariant. In fact, the set  $\text{Conv}\{\{\cup_{t=0}^{T^j} x_t^1\} \cup \mathcal{O}\}$  does not necessarily contain the  $k$ -steps robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_0^1, \pi_t^{1,*})$  from the starting state  $x_0^1$ , as shown in Figure 1.

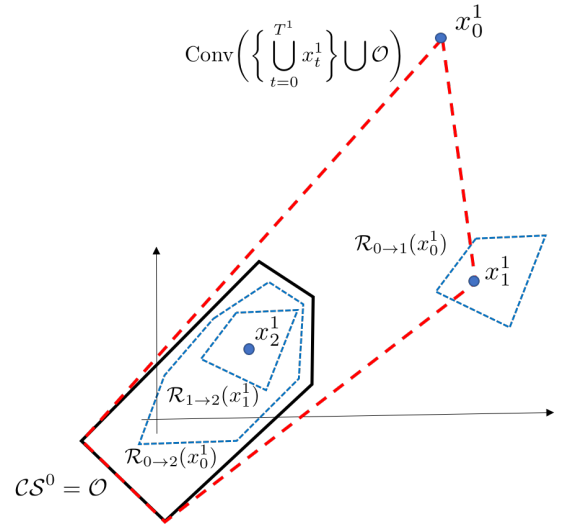


Fig. 1. The figure shows the robust invariant set  $\mathcal{O}$  (solid black), the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_0^j, \pi_t^{1,*})$  (dashed blue line) and the convex hull of the states in (9) and  $\mathcal{O}$ , denoted as  $\text{Conv}\{\{\cup_{t=0}^{T^j} x_t^1\} \cup \mathcal{O}\}$  (dashed red line). We notice that the set  $\text{Conv}\{\{\cup_{t=0}^{T^j} x_t^1\} \cup \mathcal{O}\}$  (dashed red line) does not contain the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_0^j, \pi_t^{1,*})$  (dashed blue line) and therefore it is not a robust invariant for the closed-loop system (8).

Now, we notice that robust control invariant sets can be computed using  $k$ -steps robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_0^1, \pi_t^{1,*})$  from the stored states in (9). In particular, we notice that the union of the  $k$ -steps robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_0^1, \pi_t^{1,*})$  for  $k = 0, \dots, N$  and the robust convex safe set  $\mathcal{CS}^0 = \mathcal{O}$  is a robust control invariant. Therefore, we define the *robust convex safe set* at iteration  $j=1$  as the

convex hull of the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_t^1, \pi_t^{1,*})$  and the robust convex safe set  $\mathcal{CS}^0$  at iteration 0:

$$\mathcal{CS}^1 = \text{Conv} \left( \left\{ \bigcup_{t=0}^{T^1} \bigcup_{k=0}^N \mathcal{R}_{t \rightarrow t+k}(x_t^1, \pi_t^{1,*}) \right\} \cup \mathcal{CS}^0 \right). \quad (10)$$

The above robust convex safe set at iteration  $j = 1$  is constructed for the closed-loop system (8) and it is shown in Figure 2.

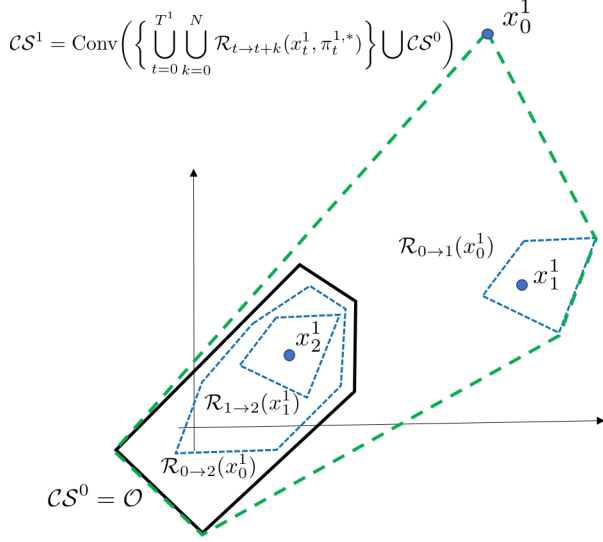


Fig. 2. The figure shows the robust invariant set  $\mathcal{O}$  (solid black line), the convex safe set  $\mathcal{CS}^1 = \text{Conv}(\{\bigcup_{t=0}^{T^1} \bigcup_{k=0}^N \mathcal{R}_{t \rightarrow t+k}(x_t^1, \pi_t^{1,*})\} \cup \mathcal{CS}^0)$  (dashed green line) and the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_t^1, \pi_t^{1,*})$  (dashed blue line).

The above process is repeated at each iteration  $j \geq 1$ . Clearly, Assumption 4 must hold when  $\mathcal{CS}^0$  is replaced with  $\mathcal{CS}^{j-1}$  and iteration 1 with  $j$ . More formally, given the  $N$ -steps policy

$$\pi_t^{j,*}(\cdot) = [\pi_{t|t}^{j,*}(\cdot), \dots, \pi_{t+N-1|t}^{j,*}(\cdot)] \quad (11)$$

and the closed-loop system

$$x_{t+1}^j = Ax_t^j + B\pi_{t|t}^{j,*}(x_t^j) + w_t^j \quad (12)$$

we assume that the following holds.

**Assumption 5:** For all  $t \in \{0, \dots, T^j\}$ , the  $N$ -steps policy  $\pi_t^{j,*}$  from (11) robustly steers the predicted closed-loop system

$$x_{k+1|t}^j = Ax_{k|t}^j + B\pi_{k|t}^{j,*}(x_{k|t}^j) + w_{k|t}^j, \quad \forall k = t, \dots, t+N-1$$

from the state  $x_t^j \in \mathcal{C}^j$  to the robust convex safe set  $\mathcal{CS}^{j-1}$  in  $N$ -steps, while robustly satisfying state and input constraints (2).

Later in Section VI, we will show how to synthesize a control policy  $\pi_t^{j,*}$  which satisfies Assumption 5.

At iteration  $j$ , we iteratively define the *robust convex safe set*:

$$\mathcal{CS}^j = \text{Conv} \left( \left\{ \bigcup_{t=0}^{T^j} \bigcup_{k=0}^N \mathcal{R}_{t \rightarrow t+k}(x_t^j, \pi_t^{j,*}) \right\} \cup \mathcal{CS}^{j-1} \right). \quad (13)$$

Details on the computation and storage of the convex safe set are provided next.

### B. Robust Convex Safe Set: Vertex Representation

Recall from Assumption 2 that  $l$  denoted the number of vertices of the disturbance support. Now, we define the  $l^k$  vertices of the  $k$ -step robust reachable set  $\mathcal{R}_{t \rightarrow t+k}(x_t^j, \pi_t^{j,*})$  from  $x_t^j$  as

$$[v_{t+k|t}^{j,1}, \dots, v_{t+k|t}^{j,l^k}]. \quad (14)$$

The vertices of the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_t^j, \pi_t^{j,*})$  for all  $k \in \{0, \dots, N-1\}$ ,  $i \in \{0, \dots, j\}$  and  $t \in \{0, \dots, T^j\}$  are collected by the following matrix

$$\mathbf{X}^j = [\mathbf{X}^{j-1}, v_{0|0}^{j,1}, \dots, v_{N-1|0}^{j,l^{N-1}}, \dots, v_{t|t}^{j,1}, \dots, v_{t+N-1|t}^{j,l^{N-1}}, \dots, v_{T^j|T^j}^{j,1}, \dots, v_{T^j+N-1|T^j}^{j,l^{N-1}}], \quad (15)$$

where at the  $j$ th iteration  $v_{t+k|t}^{j,i}$  represents the  $i$ th vertex of the robust reachable set  $\mathcal{R}_{t \rightarrow t+k}(x_t^j, \pi_t^{j,*})$ . In the above recursive definition, we set  $\mathbf{X}^0 = [v_o^1, \dots, v_o^m]$ , where  $v_o^i$  for  $i \in \{1, \dots, m\}$  are the vertices of  $\mathcal{O}$  from Assumption 2. The matrix  $\mathbf{X}^j \in \mathbb{R}^{n \times \text{col}(\mathbf{X}^j)}$ , where the number of columns  $\text{col}(\mathbf{X}^j) = (T^j + 1) \sum_{k=0}^{N-1} l^k + \text{col}(\mathbf{X}^{j-1})$ .

Finally, as the columns of the matrix  $\mathbf{X}^j$  in (15) collect all vertices of the robust reachable sets  $\mathcal{R}_{t \rightarrow t+k}(x_t^j, \pi_t^{j,*})$ , the robust convex safe set  $\mathcal{CS}^j$  from (13) can be written as

$$\mathcal{CS}^j = \left\{ x \in \mathbb{R}^n : \exists \lambda^j \geq 0, \mathbf{X}^j \lambda^j = x \text{ and } \mathbf{1}^\top \lambda^j = 1 \right\}, \quad (16)$$

where  $\mathbf{1} \in \mathbb{R}^{\text{col}(\mathbf{X}^j)}$  is a vector of ones.

**Remark 5:** Notice that the strategy presented in this paper is based on a commonly used “vertex enumeration approach” [47], [54]. Its worst case complexity is exponential in the horizon  $N$  of the feedback policy (11), although independent on the length of the task duration  $T^j$ . We underline that this paper focuses on the fundamental properties of the controller design. Computational tractability can be obtained as in any MPC scheme by using a different disturbance model or feedback parametrization.

### C. Robust Q-Function

The robust  $Q$ -function approximates the cost-to-go over the robust convex safe set and it is constructed iteratively as explained next. At iteration  $j$ , we assume that we are given the robust  $Q$ -function  $Q^{j-1}(\cdot)$  which maps each state  $x \in \mathcal{CS}^{j-1}$  to the closed-loop cost, and we show how to construct a robust  $Q$ -function at the next iteration  $j$ . This recursion is initialized at iteration 0 setting the robust  $Q$ -function

$$Q^0(x) = 0, \quad \forall x \in \mathcal{CS}^0 = \mathcal{O}. \quad (17)$$

While in the nominal case from [40] the vertices of the convex safe set are a subset of the stored trajectory, the convex safe set from (13) may introduce additional vertices representing the worst case predicted realizations. For this reason, a cost-to-go associated with such predicted worst case

realizations should be defined. In the following we define the cost-to-go  $J_{t|t}^j$  associated with the stored states  $x_{t|t}^j = x_t^j$  and the predicted cost-to-go  $J_{k|t}^j$  associated with the predicted state  $x_{k|t}^j$  at time  $k$ . In particular, after completion of the iteration  $j$  for  $t \in \{0, \dots, T^j\}$ ,  $k \in \{0, \dots, N-1\}$  and  $i \in \{1, \dots, l^k\}$ , we compute the cost-to-go for the vertices  $v_{k|t}^{j,i}$  of  $\mathcal{CS}^j$  from  $\mathbf{X}^j$  in (15) as

$$\begin{aligned} J_{k|t}^j(v_{k|t}^{j,i}) &= \min_{\gamma_k \geq 0} h(v_{k|t}^{j,i}, \pi_{k|t}^{j,*}(v_{k|t}^{j,i})) + \sum_{r=1}^{l^{k+1}} \gamma^r J_{k+1|t}^j(v_{k+1|t}^{j,r}), \\ \text{s.t. } \sum_{r=1}^{l^{k+1}} \gamma^r v_{k+1|t}^{j,r} &= A v_{k|t}^{j,i} + B \pi_{k|t}^{j,*}(v_{k|t}^{j,i}) \\ \sum_{r=1}^{l^{k+1}} \gamma^r &= 1 \end{aligned} \quad (18)$$

where  $\gamma_k = [\gamma^1, \dots, \gamma^{l^{k+1}}]$  and  $\pi_{k|t}^{j,*}(\cdot)$  is the control policy from (11). In the above recursion we set

$$J_{t+N|t}^j(v_{t+N|t}^{j,i}) = Q^{j-1}(v_{t+N|t}^{j,i}), \quad \forall i \in \{1, \dots, l^N\}. \quad (19)$$

Basically, the cost-to-go  $J_{k|t}^j(v_{k|t}^{j,i})$  at time  $k$  is computed summing up the running cost and the interpolated cost-to-go at the next predicted time  $k+1$ .

Given  $Q^{j-1}(\cdot)$ , the cost-to-go  $J_{k|t}^j(\cdot)$  is computed for all  $i \in \{0, \dots, j\}$ ,  $t \in \{0, \dots, T^j\}$  and  $k \in \{0, \dots, N-1\}$ . Then, these cost values are collected in the following vector

$$\mathbf{J}^j = [\mathbf{J}^{j-1}, J_{0|0}^j(v_{0|0}^{j,1}), \dots, J_{N-1|0}^j(v_{N-1|0}^{j,l^{N-1}}), \dots, J_{t|t}^j(v_{t|t}^{j,1}), \dots, J_{t+N-1|t}^j(v_{t+N-1|t}^{j,l^{N-1}}), \dots, J_{T^j|T^j}^j(v_{T^j|T^j}^{j,1}), \dots, J_{T^j+N-1|T^j}^j(v_{T^j+N-1|T^j}^{j,l^{N-1}})],$$

where  $\mathbf{J}^0 = [0, \dots, 0]$  represents the cost-to-go associated with the vertices of  $\mathcal{O}$ . Finally, we define the  $Q$ -function at iteration  $j$  which interpolates the cost-to-go over the robust safe set,

$$Q^j(x) = \min_{\lambda^j \in \Lambda^j(x)} \mathbf{J}^j \lambda^j, \quad (20)$$

where for the matrix  $\mathbf{X}^j$  the set

$$\Lambda^j(x) = \left\{ \lambda^j \in \mathbb{R}^{\text{col}(\mathbf{X}^j)} : \lambda^j \geq 0, \mathbf{X}^j \lambda^j = x \text{ and } \mathbf{1}^\top \lambda^j = 1 \right\} \quad (21)$$

collects the vectors  $\lambda^j$  which can be used to express  $x$  as a convex combination of the columns of  $\mathbf{X}^j$ .

*Remark 6:* Notice that a longer task duration  $T^j$  results in more data points used to compute the cost vector  $\mathbf{J}^j$ . Consequently, the  $Q$ -function from (20) can only decrease when the duration of the control task  $T^j$  is increased. As we will discuss later on, the  $Q$ -function from (20) is an upper-bound of the closed-loop performance of the certainty equivalent system, therefore the upper-bound on the cost can only decrease as the duration of the control task  $T^j$  increases.

#### D. Set of Safe Policies

At this point we have shown how to compute a robust control invariant set and a  $Q$ -function based on data collected at the  $j$ th iteration. The last missing element needed for an MPC design is the feedback controller associated with the terminal set.

Here we show how to construct a set of safe policies  $\mathcal{SP}^j$ , which may be used to robustly constrain the evolution of system (3) within  $\mathcal{CS}^j$ , while satisfying state and input constraints (2). We begin by presenting an implicit parametrization of the set of policies  $\mathcal{SP}^j$  which is amenable for optimization and it can be used to design a predictive controller that guarantees recursive constraint satisfaction. Afterwards, we define a safe policy  $\kappa^{j,*}(\cdot) \in \mathcal{SP}^j$ , which is able to complete the task from any state within the robust convex safe set  $\mathcal{CS}^j$ .

First, we define the matrix  $\mathbf{U}^j$  collecting the inputs associated with the data stored in (15),

$$\begin{aligned} \mathbf{U}^j &= [\mathbf{U}^{j-1}, \pi_{0|0}^{j,*}(v_{0|0}^{j,1}), \dots, \pi_{N-1|0}^{j,*}(v_{N-1|0}^{j,l^{N-1}}), \dots, \\ &\quad \pi_{t|t}^{j,*}(v_{t|t}^{j,1}), \dots, \pi_{t+N-1|t}^{j,*}(v_{t+N-1|t}^{j,l^{N-1}}), \dots, \\ &\quad \pi_{T^j|T^j}^{j,*}(v_{T^j|T^j}^{j,1}), \dots, \pi_{T^j+N-1|T^j}^{j,*}(v_{T^j+N-1|T^j}^{j,l^{N-1}})] \end{aligned} \quad (22)$$

where the policies  $\pi_{k|t}^{j,*}$  are defined in (11). In the above recursive definition  $\mathbf{U}^0 = [K v_0^1, \dots, K v_0^m]$ , where the feedback gain  $K$  and the vertices  $v_o^i$  of  $\mathcal{O}$  are defined in Assumption 2.

Now, we notice that by linearity of system (3), if a state  $x \in \mathcal{CS}^j$  is expressed as a convex combination of the stored states  $x = \mathbf{X}^j \lambda^j$ , then the input  $u = \mathbf{U}^j \lambda^j \in \mathcal{U}$  will keep the evolution of the system within  $\mathcal{CS}^j$  for all disturbance realizations. More formally, given the set  $\Lambda^j(\cdot)$  defined in (21), we have that  $\forall x \in \mathcal{CS}^j \subseteq \mathcal{X}, \forall \lambda^j \in \Lambda^j(x)$

$$\mathbf{U}^j \lambda^j \in \mathcal{U} \text{ and } Ax + B\mathbf{U}^j \lambda^j + w \in \mathcal{CS}^j, \forall w \in \mathcal{W}.$$

Therefore, the set of feedback policies  $\kappa^j(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^d$

$$\mathcal{SP}^j = \left\{ \kappa^j(\cdot) : \forall x \in \mathcal{CS}^j, \exists \lambda^j \in \Lambda^j(x), \text{ such that } \kappa^j(x) = \mathbf{U}^j \lambda^j \right\}, \quad (23)$$

guarantees that  $\forall \kappa^j(\cdot) \in \mathcal{SP}^j$  the robust convex safe set  $\mathcal{CS}^j$  is a robust positive invariant set for the closed-loop system  $x_{t+1} = Ax_t + B\kappa^j(x_t) + w_t$ . This statement is formalized by the following Proposition 1.

*Proposition 1:* Let Assumptions 1–5 hold. Then, for all control policies  $\kappa^j(\cdot) \in \mathcal{SP}^j$  and  $\forall x \in \mathcal{CS}^j$  we have that

$$Ax + B\kappa^j(x) + w \in \mathcal{CS}^j \subseteq \mathcal{X} \quad \forall w \in \mathcal{W}$$

and  $\kappa^j(x) \in \mathcal{U}$ .

*Proof:* The proof can be found in [55, Section XII]. ■

Finally, we define the safe policy

$$\kappa^{j,*}(x) = \mathbf{U}^j \lambda^{j,*}(x), \quad (24)$$

where  $\lambda^{j,*}(x)$  is the minimizer in (20). Basically, the above safe policy evaluated at  $x$  is given by the convex combination of stored inputs, for the multipliers  $\lambda^{j,*}(x)$  which define the robust  $Q$ -function at  $x$ . In the following propositions, we show that the  $Q$ -function is a Lyapunov function for the certainty equivalent closed-loop system (5) and (24). Furthermore, we

show that the policy (24) in closed-loop with system (3) guarantees Input-to-State Stability (ISS).

**Proposition 2:** Let Assumptions 1–5 hold. Consider the  $Q$ -function  $Q^j(\cdot)$  in (20), we have that for all  $x \in \mathcal{CS}^j$

$$Q^j(x) \geq h(x, \kappa^{j,*}(x)) + Q^j(Ax + B\kappa^{j,*}(x)), \quad (25)$$

where  $\kappa^{j,*}(\cdot)$  is the safe policy defined in (24).

*Proof:* The proof can be found in [55, Section XII]. ■

**Proposition 3:** Consider system (3) in closed-loop with the safe policy (24). Let Assumptions 1–5 hold and assume that  $x_0 \in \mathcal{CS}^j$ , then the closed-loop system (3) and (24) is input to state stable for the robust positive invariant set  $\mathcal{O}$ .

*Proof:* The proof can be found in [55, Section XII]. ■

## VI. LEARNING MODEL PREDICTIVE CONTROL

This section introduces the iterative control design procedure. At the end of iteration  $j - 1$ , we collect a data set of costs, inputs, and states, which are used to construct the robust convex safe set and robust  $Q$ -function at iteration  $j - 1$ , as described in Section V. Finally, we leverage these quantities to design a robust Learning Model Predictive Controller (LMPC) for the  $j$ th iteration. The LMPC policy is able to safely execute the control task at the  $j$ th iteration and it can be used to collect new closed-loop data to design the controller at the next iteration  $j + 1$ .

### A. Policy Synthesis

In this section, we introduce the LMPC policy. For more details on the control design choices and the differences with standard MPC design, we refer to the discussion in Section VI-B and to the properties description in Section VII.

We define the following optimal control problem for the state  $x_t^j \in \mathbb{R}^n$  and the parameter  $N_t^j \in \{0, \dots, N\}$ :

$$\begin{aligned} C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^j) = \\ \min_{\substack{M_t^j, \\ \lambda_t^j, g_t^j}} \sum_{k=t}^{t+N-1} h(\bar{x}_{k|t}^j, \pi_{k|t}^j(\bar{x}_{k|t}^j)) + Q^{j-1}(\bar{x}_{t+N|t}^j) \end{aligned} \quad (26a)$$

$$\text{s.t. } x_{t|t}^j = \bar{x}_{t|t}^j = x_t^j, \quad (26b)$$

$$\bar{x}_{k+1|t}^j = A\bar{x}_{k|t}^j + B\pi_{k|t}^j(\bar{x}_{k|t}^j), \quad (26c)$$

$$x_{k+1|t}^j = Ax_{k|t}^j + B\pi_{k|t}^j(x_{k|t}^j) + w_{k|t}^j, \quad (26d)$$

$$x_{k|t}^j \in \mathcal{X}, \quad \pi_{k|t}^j(x_{k|t}^j) \in \mathcal{U}, \quad (26e)$$

$$x_{t+N|t}^j \in \mathcal{CS}^{j-1}, \quad (26f)$$

$$\pi_{k|t}^d(x_{k|t}) = \sum_{s=0}^{k-t-1} M_{ks|t}^j w_{s|t}^j + g_{k|t}^j, \quad (26g)$$

$$\kappa_{k|t}^{j-1}(x_{k|t}^j) = \mathbf{U}^{j-1} \lambda_{k|t}^j, \quad (26h)$$

$$\lambda_{k|t}^j \in \Lambda^{j-1}(x_{k|t}^j), \quad (26i)$$

$$\pi_{i|t}^j(x_{i|t}^j) = \pi_{i|t}^d(x_{i|t}), \forall i \in \{t, \dots, t+N_t^j-1\}, \quad (26j)$$

$$\pi_{i|t}^j(x_{i|t}^j) = \kappa_{i|t}^{j-1}(x_{i|t}^j), \forall i \in \{t+N_t^j, \dots, t+N-1\}, \quad (26k)$$

$$\forall w_{k|t}^j \in \mathcal{W}, \quad \forall k = \{t, \dots, t+N-1\},$$

where the optimization variables are

$$M_t^j = \begin{bmatrix} 0 & \cdots & \\ M_{21|t} & 0 & \cdots \\ M_{31|t} & M_{32|t} & \ddots \\ \vdots & & \end{bmatrix}, \quad g_t^j = \begin{bmatrix} g_{t|t}^j \\ \vdots \\ g_{t+N-1|t}^j \end{bmatrix}$$

$\lambda_t^j = [\lambda_{t|t}^j, \dots, \lambda_{t+N-1|t}^j]$ . Equation (26k) and the parameter  $N_t^j$  describe the control policy which defines the evolution of the predicted nominal and uncertain trajectories in (26c)–(26d). In particular, for the first  $N_t^j$  predicted time steps the control policy  $\pi_{k|t}^j(\cdot)$  equals the disturbance feedback policy (26g), and for the last  $N - N_t^j$  predicted steps  $\pi_{k|t}^j(\cdot)$  equals the safe feedback policy (26h)–(26i). Equations (26b)–(26e) represent input and state constraints which must be satisfied robustly for all disturbance realizations. For  $k \in \{t, \dots, N_t^j - 1\}$ , the robust state and input constraints are reformulated using the disturbance feedback policy (26g) [50]. On the other hand, from Proposition 1, we have that for time  $k \in \{N_t^j, \dots, N\}$  the safe policy (26h) guarantees robust constraints satisfaction by construction. Finally, the terminal constraint (26f) robustly enforces  $x_{t+N|t}$  within the robust control invariant set  $\mathcal{CS}^{j-1}$ .

The finite time optimal control problem (26) is used to define the LMPC policy from Algorithm 1. Given the measured state  $x_t^j$ , Algorithm 1 solves  $N + 1$  instances of Problem (26) and it returns the optimal robust  $N$ -steps policy at time  $t$

$$\pi_t^{j,*}(\cdot) = [\pi_{t|t}^{j,*}(\cdot), \dots, \pi_{t+N-1|t}^{j,*}(\cdot)] \quad (27)$$

and the LMPC cost  $J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j)$ . Then, we apply to system (3)

$$u_t^j = \pi_{t|t}^{j,*}(x_t). \quad (28)$$

Algorithm 1 is resolved at time  $t + 1$ , based on the new state  $x_{t+1|t+1} = x_{t+1}^j$ , yielding a *moving* or *receding horizon* control strategy. Finally, the control policy (27) is used to compute the robust convex safe set  $\mathcal{CS}^j$  as discussed in Section V.

---

### Algorithm 1: LMPC Algorithm

---

- 1 **Input:** System's state  $x_t^j$
  - 2 Set  $N_t^{j,*} = \arg\min_{N_t^j \in \{0, \dots, N\}} C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^j)$
  - 3 Let  $\pi_t^{j,*}(\cdot) = [\pi_{t|t}^{j,*}(\cdot), \dots, \pi_{t+N-1|t}^{j,*}(\cdot)]$  be the optimal solution to problem  $C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^{j,*})$
  - 4 Set  $J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t) = \min_{N_t^j \in \{0, \dots, N\}} C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^j)$
  - 5 **Output:**  $\pi_t^{j,*}(\cdot)$  and  $J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j)$
- 

The above algorithm solves different instances of Problem (26) to synthesize the control policy at iteration  $j$ . In particular, in line 2 we first solve Problem (26) for all  $N_t^j \in \{0, \dots, N\}$  and then set  $N_t^{j,*}$  equal to the parameter which led to the minimum open-loop cost  $C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^{j,*})$ . Afterwards, in lines 3–4 we set the optimal control policy  $\pi_t^{j,*}(\cdot)$  and the open-loop cost  $J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t)$  equal to the optimal policy and the open-loop cost of problem  $C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^{j,*})$ , respectively.



*Remark 7:* The optimal control problem (26) computes a sequence of policies  $\pi_t^{j,*}(\cdot) = [\pi_{t|t}^{j,*}(\cdot), \dots, \pi_{t+N-1|t}^{j,*}(\cdot)]$ , which robustly steer system (3) from  $x_t^j$  to the robust convex safe set  $\mathcal{CS}^{j-1}$ . Notice that if at iteration  $j-1$  more data is collected, then the robust convex safe set  $\mathcal{CS}^{j-1}$  from (13) may be enlarged. As a result, when the tasks are defined over longer time periods  $T^j$ , the proposed iterative control design procure may synthesize control policies with larger regions of attraction.

### B. Design Choices

In standard robust MPC at each time step we solve an optimal control problem over a fixed space of feedback policies. On the other hand, in Problem (26) the space of feedback policies changes as a function of the predicted time step  $k$ . In particular, the predicted trajectory is computed using a disturbance feedback policy for  $k \leq N_t^j$  and a safe feedback policy (26h)–(26i) for  $k > N_t^j$ . In the following we discuss why this strategy allows us to guarantee recursive constraint satisfaction.

Recall that in predictive control recursive constraint satisfaction is ensured using a terminal constraint set. In particular, the terminal constraint set should be (robust) control invariant, for a feedback policy that can be used by the (robust) MPC to forecast the evolution of the system [47]. Notice that a disturbance feedback policy (or equivalently an affine state feedback policy [50]) may not be able to robustly constrain the evolution of the system within the terminal constraint set  $\mathcal{CS}^j$ . For this reason, in Problem (26) we used a time-varying feedback policy, which is defined by the parameter  $N_t^j$ , and in Algorithm 1 we solved Problem (26) for different values of  $N_t^j$ . This strategy guarantees that the safe policy can be used to robustly constrain the evolution of the predicted system within the robust safe set  $\mathcal{CS}^j$ , and it is used in Theorem 1 to show that the LMPC (26) and (28) guarantees recursive feasibility.

Furthermore, we comment on the computational tractability of the proposed strategy. As already mentioned, Algorithm 1 solves  $N+1$  instances of Problem (26) to forecast the evolution of the system using either the disturbance feedback policy or the safe policy from Section V-D. We underline that these  $N+1$  optimal control problems are independent and can be solved in parallel. Therefore, when parallel computing is available, the online computational complexity of the proposed strategy is independent on the controller horizon.

Finally, we emphasize the differences between the linear control policy defined by the feedback gain  $K$  from Assumption 2, the safe control policy  $\kappa^{j,*}(\cdot)$  in (24), and the LMPC policy (28). We notice that the linear feedback policy  $\pi^{\text{linear}}(x) = Kx$  may be used to robustly constrain the evolution of the system within the robust positive invariant set  $\mathcal{O}$ , which may be a small neighborhood of the origin. On the other hand, the safe control policy  $\kappa^{j,*}(\cdot)$  is defined on the robust convex safe set  $\mathcal{CS}^j \subseteq \mathcal{O}$  and it may be used to steer the system from any state  $x \in \mathcal{CS}^j$  to the neighborhood of the origin  $\mathcal{O}$ , as shown in the result section. Finally, the LMPC policy (28) is defined on a larger domain  $\mathcal{C}^j \subseteq \mathcal{CS}^j$  and it may be used to enlarge the robust convex safe set by collecting new

data, when the initial condition  $x_0^j$  is selected on the boundary of  $\mathcal{C}^j$ . A strategy to select such initial condition is described in Section VIII and its efficacy is demonstrated in the result Section IX-B.

## VII. PROPERTIES

This section shows that the LMPC policy (28) satisfies our design requirements from Section IV-A.

### A. Recursive Feasibility

We show that if Problem (26) is feasible at time  $t = 0$  for some  $N_0^j \in \{0, \dots, N\}$ , then the LMPC policy (28) guarantees that state and input constraints are recursively satisfied. First, we define the set

$$\mathcal{C}^j = \{x \in \mathbb{R}^n : \exists N_0^j \in \{0, \dots, N\}, C_{0 \rightarrow N}^{\text{LMPC},j}(x, N_0^j) < \infty\} \quad (29)$$

collecting the states from which Problem (26) is feasible for some  $N_0^j \in \{0, \dots, N\}$ . Next, we show that if  $x_0^j \in \mathcal{C}^j$ , then Problem (26) is feasible for all time  $t \geq 0$  and for some  $N_t^j \in \{0, \dots, N\}$ .

*Theorem 1:* Consider the closed-loop system (3) and (28). Let Assumptions 1–3 hold and  $x_0^j \in \mathcal{C}^j$ . Then, for all time  $t \geq 0$  Problem (26) is feasible for some  $N_t^j \in \{0, \dots, N\}$ , and the closed-loop system (3) and (28) satisfies state and input constraints.

*Proof:* Assume that at time  $t$  Problem (26) is feasible for some  $N_t^j \in \{0, \dots, N\}$ . At the next time  $t+1$ , by Proposition 1 we know that, for  $\kappa^{j-1,*}(\cdot) \in \mathcal{SP}^{j-1}$ , the following candidate policy

$$[\pi_{t+1|t}^{j,*}(\cdot), \dots, \pi_{t+N-1|t}^{j,*}(\cdot), \kappa^{j-1,*}(\cdot)] \quad (30)$$

is feasible for the Problem (26) for some  $N_{t+1}^j \in \{0, \dots, N\}$ . By assumption  $x_0^j \in \mathcal{C}^j$ , thus we have that Problem (26) is feasible at time  $t = 0$  for some  $N_0^j \in \{0, \dots, N\}$ . Concluding, we have shown that if Problem (26) is feasible for some  $N_t^j \in \{0, \dots, N\}$  at time  $t$ , then Problem (26) is feasible for some  $N_{t+1}^j \in \{0, \dots, N\}$  at  $t+1$ . Therefore by induction we have that for all time  $t \geq 0$  Problem (26) is feasible for some  $N_t^j \in \{0, \dots, N\}$  and the closed-loop system (3) and (28) satisfies state and input constraints (2). ■

### B. Input to State Stability (ISS)

In this section, we show that the closed-loop system (3) and (28) is ISS with respect to  $\mathcal{O}$ . We recall that in standard MPC strategies the finite time optimal control problem can be reformulated as a parametric Quadratic Program (QP). This fact is used in [50] to show continuity of the open-loop cost and then to prove ISS of the origin. In the proposed approach, the open-loop cost from Algorithm 1

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t) = \min_{N_t^j \in \{0, \dots, N\}} C_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j, N_t^j)$$

is not given by the solution to a parametric QP. Therefore, continuity cannot be guaranteed, and the standard technique from [50] cannot be used to prove ISS. Instead, we introduce the standard definition of dissipative-form ISS-Lyapunov



function for the robust invariant set  $\mathcal{O}$  [48], [52] and we show that the cost of the LMPC  $J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t)$  is an ISS-Lyapunov function.

**Definition 7:** A dissipative-form ISS-Lyapunov function for the closed-loop system (3) and (28) and the invariant set  $\mathcal{O}$  is a function  $V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  such that there exists  $\alpha_1, \alpha_2, \alpha \in \mathcal{K}_\infty$ , and  $\sigma \in \mathcal{K}$  so that for all  $x_t \in \mathbb{R}^n$  and  $w_t \in \mathbb{R}^m$ ,

$$\alpha_1(|x_t|_{\mathcal{O}}) \leq V(x_t) \leq \alpha_2(|x_t|_{\mathcal{O}}) \quad (31a)$$

$$V(Ax_t + B\pi_{t|t}^{j,*}(x_t) + w_t) - V(x_t) \leq -\alpha(|x_t|_{\mathcal{O}}) + \sigma(\|w_t\|). \quad (31b)$$

Notice that, as in [52], no assumptions on the continuity of  $V(\cdot)$  are required. However (31a) implies that  $V(\cdot)$  is continuous on the boundary of  $\mathcal{O}$ . The above definition can be used to show that the closed-loop system (3) and (28) is ISS with respect to the invariant set  $\mathcal{O}$ , as described by the following proposition.

**Proposition 4:** The following statements are equivalent:

- The closed-loop system (3) and (28) is ISS with respect to the robust invariant set  $\mathcal{O}$ .
- There exists a dissipative-form ISS-Lyapunov function  $V(\cdot)$ .

*Proof:* The proof follows from [52, Theorem 2.3] substituting  $|x|$  with  $|x|_{\mathcal{O}}$ . Note that we can replace  $|x|$  with  $|x|_{\mathcal{O}}$  as by (31a) we have that  $V(x) = 0$  iff  $|x|_{\mathcal{O}} = 0$ . ■

**Proposition 5:** Let Assumptions 1–3 hold,  $x_t^j \in \mathcal{C}^j$  and define the closed-loop system dynamics

$$f_t^j(x_t^j, w_t^j) = Ax_t + B\pi_{t|t}^{j,*}(x_t^j) + w_t^j,$$

where  $\pi_{t|t}^{j,*}(\cdot)$  is the optimal policy (27). Then there exists a constant  $L > 0$  such that

$$\begin{aligned} J_{t+1 \rightarrow t+1+N}^{\text{LMPC},j}(f_t^j(x_t^j, w_t^j)) - J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) \\ \leq -\alpha(|x_t^j|_{\mathcal{O}}) + L\|w_t^j\|, \end{aligned}$$

$\forall t \geq 0, \forall w_t^j \in \mathcal{W}$  and  $\alpha \in \mathcal{K}_\infty$ .

*Proof:* The proof can be found in [55, Section XII]. ■

The above propositions allow us to prove that the closed-loop system (3) and (28) is ISS with respect to  $\mathcal{O}$ .

**Theorem 2:** Consider the closed-loop system (3) and (28). Let Assumptions 1–3 hold and assume that  $x_0^j \in \mathcal{C}^j$ , then the closed-loop system (3) and (28) is Input-to-State Stable (ISS) for the robust positive invariant set  $\mathcal{O}$ .

*Proof:* First we show that the set  $\mathcal{O}$  is robust positive invariant for the closed-loop system (3) and (28). Assume that at time  $t$  of iteration  $j$  the state  $x_t^j \in \mathcal{O}$  and recall that the disturbance feedback policy (26g) is equivalent to state a feedback policy [50], then we have that the candidate policy

$$[\pi_{t|t}^{j,*}(x) = Kx, \dots, \pi_{t+N-1|t}^{j,*}(x) = Kx]$$

is feasible at time  $t$  of the  $j$ th iteration for  $N_t^j = N$ . Now we notice that the cost associated with the above feasible policy is zero. Therefore, we have that  $u_t^j = \pi_{t|t}^{j,*}(x_t^j) = Kx_t^j$ , which together with Assumption 2 implies that the closed-loop system  $x_{t+1}^j = Ax_t^j + B\pi_{t|t}^{j,*}(x_t^j) + w_t^j \in \mathcal{O}, \forall w_t^j \in \mathcal{W}$  and that  $\mathcal{O}$  is robust positive invariant for the closed-loop system (3)

and (28). Next, notice that for a fixed  $N_t^j$  the function  $C_{t \rightarrow t+N}^{\text{LMPC},j}(\cdot, N_t^j) : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuous at the boundary of the set  $\mathcal{O}$ . Therefore, the function  $J_{t \rightarrow t+N}^{\text{LMPC},j}(\cdot)$ , which is defined as the point-wise minimum of the functions  $C_{t \rightarrow t+N}^{\text{LMPC},j}(\cdot, N_t^j)$  for  $N_t^j \in \{0, \dots, N\}$ , is continuous at the boundary of  $\mathcal{O}$ . Continuity at the boundary of  $\mathcal{O}$ , Assumption 3 and (26a) imply the existence of  $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$  such that  $\forall x_t \in \mathcal{C}^j$

$$\alpha_1(|x_t|_{\mathcal{O}}) \leq h(x_t, 0) \leq J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t) \leq \alpha_2(|x_t|_{\mathcal{O}}). \quad (32)$$

In the above equation, we used compactness of the sets  $\mathcal{X}$  and  $\mathcal{U}$ . Finally, from Proposition 5, we have that  $\forall x_t \in \mathcal{C}^j$

$$\begin{aligned} J_{t+1 \rightarrow t+1+N}^{\text{LMPC},j}(Ax_t + B\pi_{t|t}^{j,*}(x_t) + w_t) - J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t) \\ \leq -\alpha(|x_t|_{\mathcal{O}}) + \sigma(\|w_t\|_2) \end{aligned}$$

and therefore  $J_{t \rightarrow t+N}^{\text{LMPC},j}(\cdot)$  is a ISS-Lyapunov function and the closed-loop system (3) and (28) is Input-to-State Stable for the robust positive invariant set  $\mathcal{O}$ . ■

**Remark 8:** We underline that the ISS property does not imply that the closed-loop system converges in finite time to the set  $\mathcal{O}$ . However, finite time convergence may be obtained following the approach presented in [56], if at time  $T^j$  the system has not reach  $\mathcal{O}$ . In particular, the optimal policy  $\pi_{T^j}^{j,*}(\cdot) = [\pi_{T^j|T^j}^{j,*}(\cdot), \dots, \pi_{T^j+N-1|T^j}^{j,*}(\cdot)]$  computed by the finite time optimal control problem (26) can be used to robustly steer the system to the terminal set  $\mathcal{CS}^{j-1}$ . Then, from a state  $x_{T^j+N-1}^j \in \mathcal{CS}^j$  it would be possible to solve the finite time optimal control problem (26) for  $j-1$  and leverage the time-varying optimal policy  $\pi_{T^j-1}^{j-1,*}(\cdot)$  to steer the system in finite time to  $\mathcal{CS}^{j-2}$ . This procedure may be iterated to steer the system in finite time to  $\mathcal{CS}^0 = \mathcal{O}$ . Finally, we point out that robust exponential stability of the closed-loop system may be obtained decoupling the nominal and error dynamics as shown in [42]. However, this strategy cannot be applied when using a disturbance feedback policy as in (26).

### C. Performance Bound

Finally, we show that whenever  $x_0^j \in \mathcal{CS}^{j-1}$  the robust  $Q$ -function at iteration  $j-1$  is an upper-bound to the closed-loop cost of the certainty equivalent system at the next  $j$ th iteration.

**Theorem 3:** Consider the certainty equivalent system (5) in closed-loop with the LMPC (26) and (28). Let Assumptions 1–3 hold and  $x_0^j = \bar{x}_0^j \in \mathcal{CS}^{j-1} \subseteq \mathcal{C}^j$ . Then, we have that the iteration cost of the certainty equivalent closed-loop system (5) and (28) is an upper-bound to the  $Q$ -function constructed at the previous iteration, i.e.,

$$J_{0 \rightarrow T^j}^j(\bar{x}_0^j) = \sum_{t=0}^{T^j} h(\bar{x}_t^j, \pi_{t|t}^{j,*}(\bar{x}_t^j)) \leq Q^{j-1}(\bar{x}_0^j) \quad (33)$$

where  $\pi_{t|t}^{j,*}(\cdot)$  is the optimal control policy in (27).

*Proof:* By Proposition 2 we have that

$$\begin{aligned}
 Q^{j-1}(\bar{x}_0^j) &\geq h(\bar{x}_0^j, \kappa^{j-1,*}(\bar{x}_0^j)) + Q^{j-1}(\bar{x}_1^j) \\
 &\geq h(\bar{x}_0^j, \kappa^{j-1,*}(\bar{x}_0^j)) + h(\bar{x}_1^j, \kappa^{j-1,*}(\bar{x}_1^j)) \\
 &\quad + Q^{j-1}(\bar{x}_2^j) \\
 &\geq \sum_{k=0}^{N-1} h(\bar{x}_k^j, \kappa^{j-1,*}(\bar{x}_k^j)) + Q^{j-1}(\bar{x}_N^j) \\
 &\geq J_{0 \rightarrow N}^{\text{LMPC},j}(\bar{x}_0^j),
 \end{aligned} \tag{34}$$

where the last inequality holds by the feasibility of the safe policy from Section V-D for  $x_0^j \in \mathcal{CS}^{j-1} \subseteq \mathcal{C}^j$ .

Now consider the LMPC cost at time  $t$ , by Proposition 2 we have that

$$\begin{aligned}
 J_{t \rightarrow t+N}^{\text{LMPC},j}(\bar{x}_t^j) &= \sum_{k=t}^{t+N-1} h(\bar{x}_{k|t}^{j,*}, \pi_{k|t}^{j,*}(\bar{x}_{k|t}^{j,*})) + Q^{j-1}(\bar{x}_{t+N|t}^{j,*}) \\
 &\geq \sum_{k=t}^{t+N-1} h(\bar{x}_{k|t}^{j,*}, \pi_{k|t}^{j,*}(\bar{x}_{k|t}^{j,*})) + h(\bar{x}_{t+N|t}^{j,*}, \kappa^{j-1,*}(\bar{x}_{t+N|t}^{j,*})) \\
 &\quad + Q^{j-1}(A\bar{x}_{t+N|t}^{j,*} + B\kappa^{j-1,*}(\bar{x}_{t+N|t}^{j,*})) \\
 &= h(\bar{x}_{t|t}^{j,*}, \pi_{t|t}^{j,*}(\bar{x}_{t|t}^{j,*})) + \sum_{k=t+1}^{t+N-1} h(\bar{x}_{k|t}^{j,*}, \pi_{k|t}^{j,*}(\bar{x}_{k|t}^{j,*})) \\
 &\quad + h(\bar{x}_{t+N|t}^{j,*}, \kappa^{j-1,*}(\bar{x}_{t+N|t}^{j,*})) \\
 &\quad + Q^{j-1}(A\bar{x}_{t+N|t}^{j,*} + B\kappa^{j-1,*}(\bar{x}_{t+N|t}^{j,*})) \\
 &\geq h(\bar{x}_t^j, \bar{u}_t^j) + J_{t+1 \rightarrow t+1+N}^{\text{LMPC},j}(A\bar{x}_t^j + B\bar{u}_t^j),
 \end{aligned}$$

where  $\bar{u}_t^j = \pi_{t|t}^{j,*}(\bar{x}_t^j)$ . The above equation implies that the LMPC cost is non-increasing and for the closed-loop trajectory of the certainty equivalent we have

$$J_{t+1 \rightarrow t+1+N}^{\text{LMPC},j}(\bar{x}_{t+1}^j) - J_{t \rightarrow t+N}^{\text{LMPC},j}(\bar{x}_t^j) \leq -h(\bar{x}_t^j, \bar{u}_t^j),$$

which in turns implies that

$$\begin{aligned}
 J_{0 \rightarrow N}^{\text{LMPC},j}(\bar{x}_0^j) &\geq \sum_{k=0}^{T^j} h(\bar{x}_k^j, \bar{u}_k^j) + J_{T^j+1 \rightarrow T^j+1+N}^{\text{LMPC},j}(\bar{x}_{T^j+1}^j) \\
 &\geq \sum_{k=0}^{T^j} h(\bar{x}_k^j, \bar{u}_k^j) = J_{0 \rightarrow T^j}^j(\bar{x}_0^j).
 \end{aligned}$$

Finally, from the above equation and (34) we conclude that

$$Q^{j-1}(\bar{x}_0^j) \geq J_{0 \rightarrow N}^{\text{LMPC},j}(\bar{x}_0^j) \geq J_{0 \rightarrow T^j}^j(\bar{x}_0^j).$$

## VIII. REGION OF ATTRACTION APPROXIMATION

In this section, we present an algorithm which may be used to select the initial condition  $x_0^j$  or to approximate the region of attraction of the LMPC policy  $\mathcal{C}^j$  in (29). We underline that the region of attraction  $\mathcal{C}^j$  may be computed solving problem (26) as a set of parametric optimization problems with parameter  $x_t^j$  [47]. However, this computation may be prohibitive and therefore we propose a strategy to compute an inner approximation to the region of attraction  $\mathcal{C}^j$ .

Given a vector  $d \in \mathbb{R}^n$ , we define the following optimization problem:

$$\begin{aligned}
 P(N_t^j, d) &= \min_{x_0, \mathcal{M}_t^j, \lambda_t^j, g_t^j} d^\top x_0 \\
 \text{s.t.} \quad &(d^\perp)^\top x_0 = 0 \\
 &x_{t|t}^j = \bar{x}_{t|t}^j = x_0 \\
 &(26c) - (26k) \\
 &\forall w_{k|t}^j \in \mathcal{W}, \forall k = \{t, \dots, t+N-1\},
 \end{aligned} \tag{35}$$

where  $d^\perp \in \mathbb{R}^n$  is a vector perpendicular to  $d \in \mathbb{R}^n$ . Basically, the above optimization problem finds the optimal initial state  $x_0^*$ , which is farthest from the origin along the direction  $d$  and guarantees that problem  $C_{t \rightarrow t+N}^{\text{LMPC},j}(x_0, N_t^j)$  from (26) is feasible. Therefore, given a user-defined set of vectors  $\mathcal{D} = \{d^1, \dots, d^k\}$ , problem (35) can be solved repeatedly to approximate the region of attraction  $\mathcal{C}^j$ . In particular, for each vector  $d^i \in \mathcal{D}$  and parameter  $N_t^j \in \{1, \dots, N\}$  we solve problem  $P(N_t^j, d^i)$ . Afterwards, we define the approximated region of attraction  $\tilde{\mathcal{C}}^j$  as the convex hull of the optimal initial states  $x_0^*$  associated with different  $N_t^j$  and vector  $d^i$ . Algorithm 2 summaries this procedure.

---

### Algorithm 2: Region of Attraction Approximation

---

```

1 Input: Set of vectors  $\mathcal{D} = \{d^1, \dots, d^k\}$  and horizon  $N$ 
2 Initialize  $\tilde{\mathcal{C}}^j = \emptyset$ 
3 for  $d^i \in \mathcal{D}$  do
4   for  $N_t^j \in \{1, \dots, N\}$  do
5     Solve  $P(N_t^j, d^i)$  from (35)
6     Let  $x_0^*$  be the optimal initial state from  $P(N_t^j, d^i)$ 
7     Set  $\tilde{\mathcal{C}}^j = \text{Conv}\{\tilde{\mathcal{C}}^j \cup \{x_0^*\}\}$ 
8   end
9 end
10 Output: Approximate region of attraction  $\tilde{\mathcal{C}}^j$ 

```

---

Finally, we underline that the optimization problem (35) may be used to select the initial condition at iteration  $j$ . In particular given a direction  $d^j$ , the initial condition  $x_0^j$  may be chosen as the optimal initial state  $x_0^*$  for problem  $P(N, d^j)$ , as shown in the result section.

## IX. SIMULATION RESULTS

We test the proposed controller on a system subject to bounded additive uncertainty. First, we show that the proposed strategy is able to improve the performance of a system executing an iterative task. Afterwards, we show that the proposed LMPC can be used to iteratively construct a robust convex safe set  $\mathcal{CS}^j$ , which is defined over progressively larger regions of the state space. Finally, we show that data collected by the LMPC can be exploited to construct the safe policy (24), which robustly steers the uncertain system from any state within the robust safe set  $\mathcal{CS}^j$  to the set  $\mathcal{O}$ .

We consider the following double integrator system

$$x_{t+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t + w_t,$$

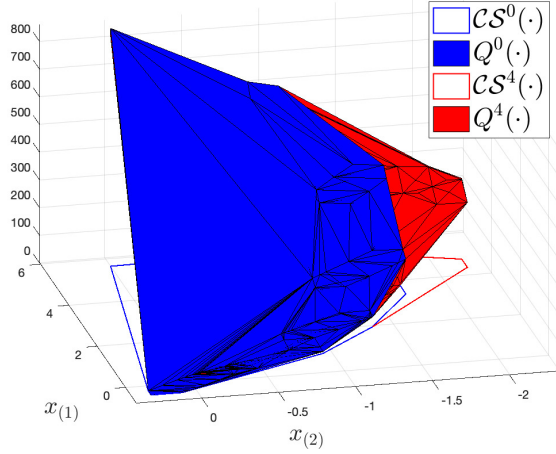


Fig. 3. Comparison between the robust safe set and  $Q$ -function at the first and last iteration.

where  $w_t \in \{w \in \mathbb{R}^2 : \|w\|_\infty \leq 0.1\}$ , subject to the following constraints  $x_t \in \mathcal{X} = \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 10\}$  and  $u_t \in \mathcal{U} = \{u \in \mathbb{R} : \|u\|_\infty \leq 1\}$  for all time instant  $t \geq 0$ . Furthermore, we define the running cost  $h(x, u) = 10|x|_{\mathcal{O}} + |u|_{K\mathcal{O}}$ , which satisfies Assumption 3.

#### A. Iterative Task

We use the LMPC (26) and (28) to iteratively steer the system from  $x_0 = [5.656; 0]$  to the robust invariant set  $\mathcal{O}$ . We designed a sub-optimal robust MPC to perform the 0th iteration and to construct the robust safe set  $\mathcal{CS}^0$  and the robust  $Q$ -function  $Q^0(\cdot)$ , which are used to initialize the LMPC with  $N = 3$ . We underline that the closed-loop performance of the robust MPC can be improved tuning its parameters. The goal of this section is to show that, given a safe sub-optimal policy, the proposed strategy may be used to iteratively improve the closed-loop performance. In the next section, we will show that the proposed strategy allows us also to enlarge the region of attraction associated with the safe sub-optimal policy used to initialize the algorithm.

We perform 4 iterations of the control task for the certainty equivalent system. At each  $j$ th iteration, we store the LMPC predicted policy and the closed-loop data in order to construct the robust safe set  $\mathcal{CS}^j$  and the robust  $Q$ -function  $Q^j(\cdot)$ . Table I shows that the closed-loop cost of the certainty equivalent system decreases, until it converges to a steady state value after 4 iterations.

TABLE I  
CLOSED-LOOP COST  $J_{0 \rightarrow T_j}^j(x_0)$  FOR ITERATION  $i \in \{0, \dots, 4\}$ .

$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
863.4245	827.9588	827.9380	827.9371	827.9371

Finally, in Figure 3 we compare the robust safe set and the robust  $Q$ -function at the first and last iteration. First, we notice that the robust safe set, which represents the domain of the  $Q$ -function, is enlarged. Furthermore, we confirm that

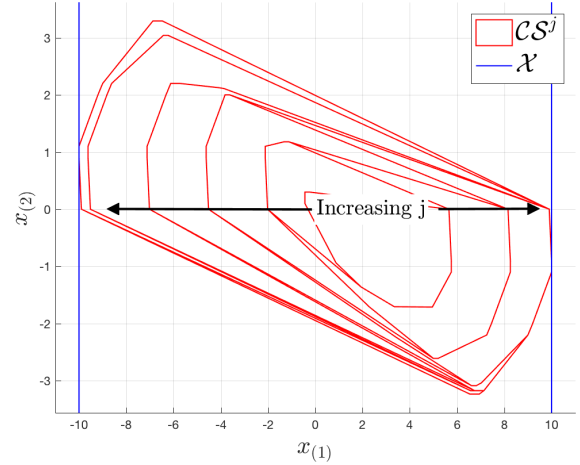


Fig. 4. Evolution of the robust safe set at each iteration.

$Q^j(\cdot)$  is non-increasing (i.e.  $Q^0(x) \leq Q^4(x), \forall x \in \mathcal{CS}^4$ ) and therefore it guarantees better bounds on the performance of the certainty equivalent closed-loop system (5) and (28), as shown in Theorem 3.

#### B. Domain Enlargement

We show that the domain of the LMPC policy may be iteratively enlarged. At each iteration, we simulate the uncertain closed-loop system (3) and (28) and we store both the closed-loop data and the predicted LMPC policy (28). The stored data are used to construct the robust safe set  $\mathcal{CS}^j$  and robust  $Q$ -function as described in Section V.

Furthermore, we use the strategy proposed in Section VIII to iteratively select the initial condition. At each iteration  $j$ , we define the vector  $d^j = [(-1)^j, 0]$  and we pick  $x_0^j$  as the optimal initial state  $x_0^*$  from problem  $P(N, d^j)$  in (35). Notice that by definition (13) we have that  $\mathcal{CS}^j \subseteq \mathcal{CS}^{j+1}$ , therefore the set of states which can be steered to  $\mathcal{O}$  by the LMPC (26) and (28) does not shrink, i.e.,  $\mathcal{C}^j \subseteq \mathcal{C}^{j+1}$ .

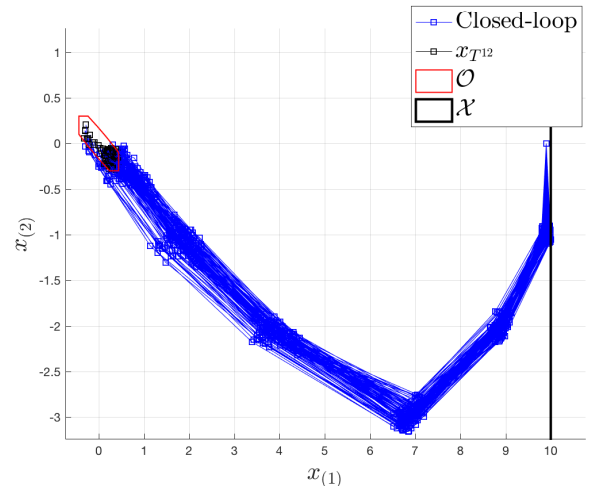


Fig. 5. Closed-loop trajectories for different disturbance realizations.

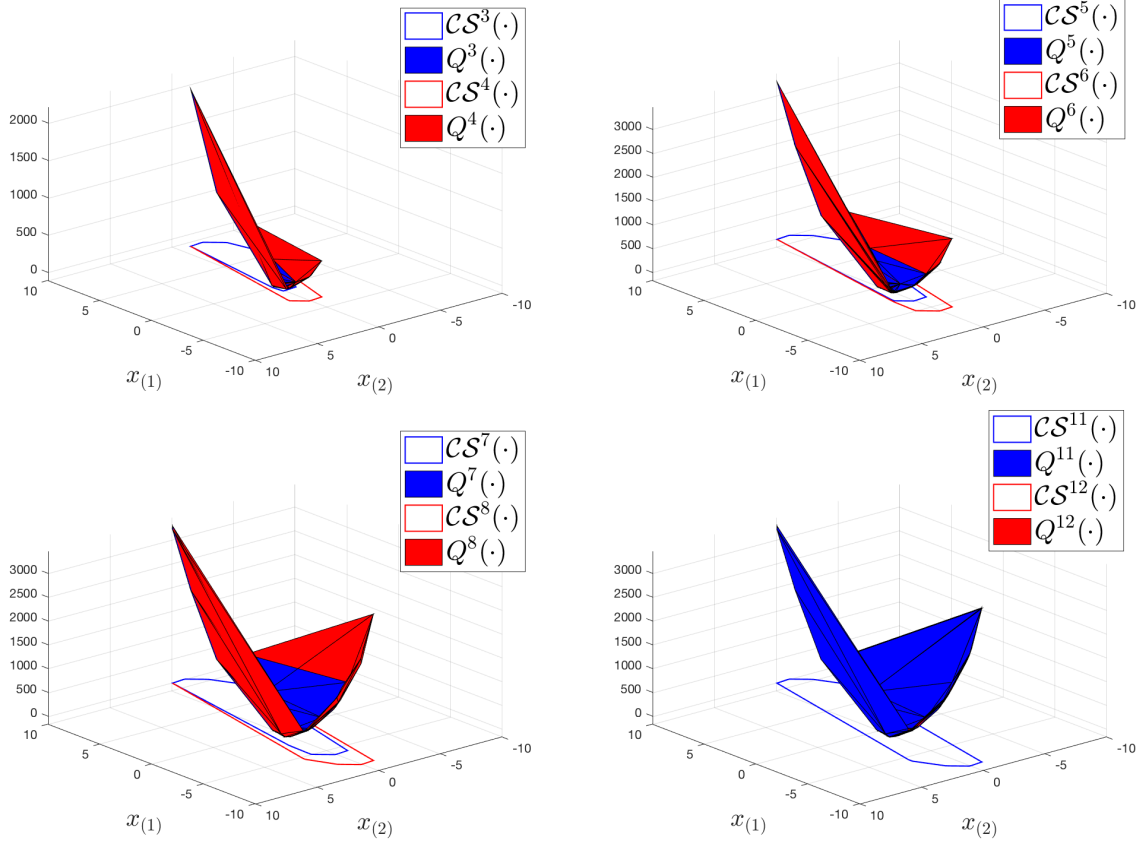


Fig. 6. Evolution of the robust  $Q$ -function  $Q^j$  at each iteration. Notice that  $Q^j(\cdot)$  (in blue) is a lower-bound to  $Q^{j+1}(\cdot)$  (in red) for all  $i \in \{3, 5, 7, 11\}$ , until convergence is reached and  $Q^{11}(\cdot) = Q^{12}(\cdot)$ .

Figure 4 shows that the robust convex safe set  $\mathcal{CS}^j$  grows at each iteration, until it converges to a set which saturates the state constraints. We underline that the closed-loop data used to enlarge  $\mathcal{CS}^j$  are generated by the LMPC, which steers the system to regions of the state space associated with low cost values. In other words, the growth of the robust safe set is cost-driven. More importantly, the iterative enlargement of the  $\mathcal{CS}^j$  is performed safely. Indeed, the LMPC guarantees robust state and input constraints satisfaction at each iteration.

Figure 5 shows 100 Monte Carlo simulations of the closed-loop system for the 12th iteration. We notice that the closed-loop trajectories satisfy state constraints and converge to the set  $\mathcal{O}$ , regardless of the disturbance realization.

Finally, Figure 6 shows the growth of the  $Q$ -function  $Q^j(\cdot)$ , which is non-increasing at each iteration. It is important to underline that  $Q^j(\cdot)$  is piece-wise affine as it is the solution to

a parametric LP [47]. Furthermore, we notice that  $Q^j(\cdot)$ , which is an upper-bound of the closed-loop cost of the disturbance-free system, resembles a quadratic function. This result is expected as the optimal value function for this problem is piece-wise quadratic [47].

### C. Exploiting the safe policy

In this section, we use the stored data from Section V-D to construct the safe policy (24). We tested this policy for 1000 Monte Carlo simulations, where we randomly sampled the initial condition  $x_0$  from the robust convex safe set  $\mathcal{CS}^{12}$ . We confirm that, for all initial conditions  $x_0 \in \mathcal{CS}^{12}$  and disturbance realizations, the safe policy (24) steered the system to the set  $\mathcal{O}$ , as shown in Figure 7.

We compare the performance of the uncertain system (3) in closed-loop with the safe policy (24) and the LMPC (26) and (28). In particular, we simulated both the closed-loop system (3) and (24) and the closed-loop system (3) and (28) for the same random initial condition  $x_0 \in \mathcal{CS}^{12}$  and disturbance realization. As reported in Table II, on average it takes  $\sim 5$ ms to evaluate the safe policy (24) and  $\sim 4.6$ s to evaluate the LMPC policy (28). This result is expected as the safe policy (24) is evaluated solving a LP and the LMPC policy (28) solving  $N + 1 = 4$  QPs. On the other hand, it

TABLE II  
PERFORMANCE OF THE LMPC POLICY (28) AND THE SAFE POLICY (24) IN CLOSED-LOOP WITH THE UNCERTAIN SYSTEM (3).

	Average solver time	Average closed-loop cost
LMPC	4.6s	77.1
Safe Policy	5ms	80

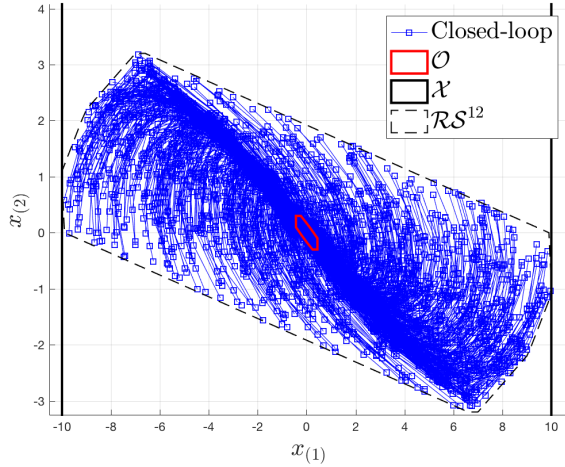


Fig. 7. Closed-loop trajectories for different disturbance realizations and initial conditions.

is interesting to notice that the closed-loop cost associated with the safe policy (24) is on average  $\sim 3\%$  higher than the cost associated with the LMPC policy (28), as shown in Table II. This result suggests that, in applications where the computational power is not always available, one can first use the proposed LMPC to iteratively construct a large robust convex safe set  $\mathcal{CS}^j$  and robust  $Q$ -function  $Q^j$ . Afterwards, these quantities can be leveraged to synthesize a safe control policy, which at the cost of slightly worse performance is able to reduce the computational burden.

## X. CONCLUSIONS

In this work we proposed a robust Learning Model Predictive Controller (LMPC) for linear systems subject to bounded additive uncertainty. At each execution of the control task, we store both the closed-loop data and the optimal predicted policy of the LMPC. First, we show how the stored data can be combined with the optimal policy from the LMPC to construct a safe set and an approximation to the value function. Afterwards, we design a safe policy which may be used to complete the task from any state in the safe set. Finally, the safe set, the value function approximation, and the safe policy are used in the control design, which guarantees input to state stability, robust constraint satisfaction, and performance bounds for the certainty equivalent closed-loop system. The effectiveness of the proposed LMPC is tested on a numerical example.

## XI. ACKNOWLEDGMENT

The authors would like to thank Monimoy Bujarbaruah and Siddharth Nair for helpful discussions and reviews. Some of the research described in this review was funded by the Hyundai Center of Excellence at the University of California, Berkeley. This work was also sponsored by the Office of Naval Research. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements,

either expressed or implied, of the Office of Naval Research or the US government.

## REFERENCES

- [1] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 259–286, 2018.
- [2] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2019.
- [3] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.
- [4] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proceedings of the 2004 American Control Conference*, vol. 3. IEEE, 2004, pp. 2214–2219.
- [5] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6059–6066.
- [6] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 491–496.
- [7] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmPC with gaussian process dynamics for autonomous miniature race cars," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1341–1348.
- [8] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, 2019.
- [9] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini, "Learning multi-step prediction models for receding horizon control," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1335–1340.
- [10] M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis, "General interpolation in mpc and its advantages," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1092–1096, 2003.
- [11] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4029–4036.
- [12] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based nmPC enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.
- [13] F. D. Brunner, M. Lazar, and F. Allgöwer, "Stabilizing linear model predictive control: On the enlargement of the terminal set," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 511–517.
- [14] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: a predictive control approach," *IEEE Transactions on Control Systems Technology*, 2019.
- [15] K. S. Lee and J. H. Lee, "Model predictive control for nonlinear batch processes with asymptotically perfect tracking," *Computers and Chemical Engineering*, vol. 21, pp. S873–S879, 1997.
- [16] K. S. Lee and J. H. Lee, "Convergence of constrained model-based predictive control for batch processes," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1928–1932, 2000.
- [17] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.
- [18] X. Lu and M. Cannon, "Robust adaptive tube model predictive control," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 3695–3701.
- [19] M. Lorenzen, M. Cannon, and F. Allgöwer, "Robust mpc with recursive model update," *Automatica*, vol. 103, pp. 461–471, 2019.
- [20] M. Lorenzen, F. Allgöwer, and M. Cannon, "Adaptive model predictive control with robust constraint satisfaction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3313–3318, 2017.
- [21] M. Tanaskovic, L. Fagiano, R. Smith, P. Goulart, and M. Morari, "Adaptive model predictive control for constrained linear systems," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 382–387.
- [22] M. Tanaskovic, L. Fagiano, R. Smith, and M. Morari, "Adaptive receding horizon control for constrained mimo systems," *Automatica*, vol. 50, no. 12, pp. 3019–3029, 2014.



- [23] M. Bujarbaruah, X. Zhang, U. Rosolia, and F. Borrelli, "Adaptive MPC for iterative tasks," *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6322–6327, 2018.
- [24] M. Bujarbaruah, X. Zhang, and F. Borrelli, "Adaptive MPC with chance constraints for FIR systems," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 2312–2317.
- [25] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli, "Adaptive MPC under time varying uncertainty: Robust and Stochastic," *arXiv preprint arXiv:1909.13473*, 2019.
- [26] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific Belmont, MA, 1996, vol. 5.
- [27] D. P. Bertsekas, "Feature-based aggregation and deep reinforcement learning: a survey and some new implementations," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 1–31, 2019.
- [28] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.
- [29] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.
- [30] S. J. Bradtko and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine learning*, vol. 22, no. 1-3, pp. 33–57, 1996.
- [31] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 1999, pp. 207–226.
- [32] J. Fleming, B. Kouvaritakis, and M. Cannon, "Robust tube mpc for linear systems with multiplicative uncertainty," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1087–1092, 2014.
- [33] M. Evans, M. Cannon, and B. Kouvaritakis, "Robust and stochastic linear mpc for systems subject to multiplicative uncertainty," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 335–341, 2012.
- [34] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, 2018.
- [35] S. Kaynama, M. Oishi, I. M. Mitchell, and G. A. Dumont, "The continual reachability set and its computation using maximal reachability techniques," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 6110–6115.
- [36] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.
- [37] A. Liniger and J. Lygeros, "Real-time control for autonomous racing based on viability theory," *IEEE Transactions on Control Systems Technology*, no. 99, pp. 1–15, 2017.
- [38] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7130–7135.
- [39] F. Blanchini and F. A. Pellegrino, "Relatively optimal control: the static solution," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 676–681, 2005.
- [40] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks: a computationally efficient approach for linear system," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3142–3147, 2017.
- [41] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, 2017.
- [42] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.
- [43] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.
- [44] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Transactions on automatic control*, vol. 48, no. 9, pp. 1600–1606, 2003.
- [45] H. Chen, C. Scherer, and F. Allgöwer, "A robust model predictive control scheme for constrained linear systems," *IFAC Proceedings Volumes*, vol. 31, no. 11, pp. 61–66, 1998.
- [46] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model predictive control for iterative tasks: Learning from experience," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1157–1162.
- [47] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for linear and hybrid systems*. Cambridge University Press, 2017.
- [48] Y. Lin, E. Sontag, and Y. Wang, "Various results concerning set input-to-state stability," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 2. IEEE, 1995, pp. 1330–1335.
- [49] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.
- [50] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.
- [51] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.
- [52] L. Grüne and C. M. Kellett, "Iss-lyapunov functions for discontinuous discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3098–3103, 2014.
- [53] C. M. Kellett, "A compendium of comparison function results," *Mathematics of Control, Signals, and Systems*, vol. 26, no. 3, pp. 339–374, 2014.
- [54] P. O. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic control*, vol. 43, no. 8, pp. 1136–1142, 1998.
- [55] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model predictive control for linear systems performing iterative tasks," *arXiv preprint arXiv:1911.09234*, 2019.
- [56] W. Langson, I. Chrysoschoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.



**Ugo Rosolia** received the B.S. and M.S. (*cum laude*) degrees from the Politecnico di Milano, Milan, Italy, in 2012 and 2014, respectively. In 2019, he received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA. He is currently a post-doctoral fellow at Caltech.

He was a Visiting Scholar with Tongji University, Shanghai, China, for the Double Degree Program PoliTong from 2010 to 2011. During his M.S. degree, he was a Visiting Student for two semesters with the University of Illinois at Urbana-Champaign, Urbana, IL, USA, sponsored by a Global E3 Scholarship. He was a Research Engineer with Siemens PLM Software, Leuven, Belgium, in 2015, where he was involved in the optimal control algorithms. His current research interests include nonlinear optimal control for the centralized and decentralized system, the iterative learning control, and the predictive control.



**Xiaojing Zhang** received his B.Sc and M.Sc. degrees in electrical engineering and information technology, in 2010 and 2012, respectively, from the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. In 2016, he received his Ph.D. degree in automatic control, also from ETH Zurich. He is currently a post-doctoral fellow and the Associate Director at the Hyundai Center of Excellence at the University of California, Berkeley. His research interests include modeling, analysis, and control of stochastic uncertain systems, randomized and data-driven algorithms, as well as stochastic optimization, with application to autonomous vehicles, energy-efficient buildings and electric power systems.



**Francesco Borrelli** received the Laurea degree in computer science engineering from the University of Naples Federico II, Naples, Italy, in 1998, and the Ph.D. degree from ETH-Zurich, Zurich, Switzerland, in 2002.

He is currently an Associate Professor with the Department of Mechanical Engineering, University of California, Berkeley, CA, USA. He is the author of more than 100 publications in the field of predictive control and author of the book *Constrained Optimal Control of Linear and Hybrid Systems* (Springer Verlag). His research interests include constrained optimal control, model predictive control and its application to advanced automotive control and energy efficient building operation.

Dr. Borrelli was the recipient of the 2009 National Science Foundation CAREER Award and the 2012 IEEE Control System Technology Award. In 2008, he was appointed the chair of the IEEE technical committee on automotive control.