Sample-Based Learning Model Predictive Control for Linear Uncertain Systems

Ugo Rosolia and Francesco Borrelli

Abstract—We present a sample-based Learning Model Predictive Controller (LMPC) for constrained uncertain linear systems subject to bounded additive disturbances. The proposed controller builds on earlier work on LMPC for deterministic systems. First, we introduce the design of the safe set and value function used to guarantee safety and performance improvement. Afterwards, we show how these quantities can be approximated using noisy historical data. The effectiveness of the proposed approach is demonstrated on a numerical example. We show that the proposed LMPC is able to safely explore the state space and to iteratively improve the worst-case closed-loop performance, while robustly satisfying state and input constraints.

I. INTRODUCTION

Exploiting historical data in order to iteratively improve the performance of Model Predictive Controllers (MPC) has been an active theme of research in the past few decades [1]–[11]. The key idea is to use stored state-input pairs in order to compute at least one of the following three components used in the control design: i) a model which describes the evolution of the system, ii) a safe set of states (and an associated control policy $\pi(\cdot)$) from which the control task can be safely completed and iii) a value function which represents the cumulative cost from a given point of the safe set when the policy $\pi(\cdot)$ is used. In this work, we focus on a strategy to build a safe set and the associated value function by exploiting historical noisy closed-loop trajectories.

Strategies to approximate the value function are studied in Approximate Dynamic Programming (ADP) [12]–[16]. In particular, ADP aims to address the curse of dimensionality associated with the computation of the optimal control policy, by exploiting a value function estimate in the controller design. This estimate may be computed using data-based or model-based strategies. A complete survey goes beyond the scope of this work, we refer the reader to [13] for further details.

The integration of MPC with system identification strategies has been extensively studied in the literature [1]–[7]. In [5] the authors identified the system’s model using a deep neural network, which incorporates uncertainty using an ensemble of models. Another system identification strategy consists of fitting a Gaussian Process (GP) to experimental data [2]–[4]. GP provides a nominal model and confidence bounds, which may be used to tighten the constraint set over the planning horizon. This strategy allows to provide high-probability safety guarantees [3], [4]. The effectiveness of GP-based strategies on experimental platform has been shown in [4], where a MPC is used to race a 1/43-scale vehicle. Regression strategies may also be used to identify the system’s model [6], [7]. For instance, the authors in [6] used a linear regression strategy to identify both a nominal model and the model uncertainty used for Robust MPC design. In [7], we used local linear regression to identify the model used by the controller, which was able to drive a 1/10-scale race car at the limit of handling.

Also the computation of safe sets has been extensively studied in the literature [17]–[23]. In reachability-based strategies, safe sets are computed solving a two players game between the controller and the disturbance [17]–[19]. Furthermore, these strategies provide a control policy which can be used to guarantee safety, by robustly constraining the evolution of the system into the safe set [17]. Also, viability theory may be used to compute safe sets [20]. The authors in [20] showed how to compute an inner approximation of the viability kernel and demonstrated the effectiveness on a RC-car set-up. Recently the computation of safe sets from data gained more attention [21]–[23]. In [21] the authors showed how to compute safe sets from data. However, the associated control policy does not guarantee that the evolution of the system is robustly contained into the safe set (i.e. the safe set is not robust invariant). In [22], [23] we showed 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 work we present a sample-based Learning Model Predictive Controller (LMPC) for linear systems subject to bounded additive uncertainty. We refer to a control task execution as “iteration” and we systematically update the control policy. In particular at each $j$th iteration, we show that the $j$th LMPC policy can be used to construct the safe set. Furthermore, we define a value function which approximates the cost-to-go associated with the $j$th LMPC policy. The safe set and value function at the $j$th iteration are used to design the LMPC at next $j+1$ iteration. We show that the proposed strategy guarantees that: i) state and input constraint are robustly satisfied, ii) the closed-loop system converges asymptotically to a neighborhood of the origin, iii) the worst-case performance of the $j$th LMPC policy is non-increasing with the iteration index, and iv) the domain on which the LMPC policy is defined is not shrinking at each $j$th iteration. The proposed control strategy is computationally intensive. Therefore, we propose a practical algorithm which exploits stored noisy closed-loop trajectories in order to approximate the safe set and
the value function. The properties of these approximations, which depends on the number of stored trajectories, are analyzed in the result section.

II. PROBLEM DEFINITION

We consider the following linear time invariant system

\[ x_{k+1}^j = Ax_k^j + Bu_k^j + w_k^j \]  

where at time \( t \) of the \( j \)th iteration the disturbance \( w_k^j \in W \), the state \( x_k \in \mathbb{R}^n \) and input \( u_k^j \in \mathbb{R}^d \). Furthermore, the system is subject to the following convex polytopic state and input constraints, for all \( k \geq 0 \)

\[ x_k \in \mathcal{X} \text{ and } \pi^j(x_k^j) \in \mathcal{U}. \]

At each \( j \)th iteration, we define the worst-case iteration cost associated with the control policy \( \pi^j(\cdot) \), as the solution to the following Bellman recursion

\[ J_{\pi^j}(x_0^j) = \max_{w \in W} [h(x_0^j, u_0^j) + J_{\pi^j}(Ax_0^j + Bu_0^j + w)], \]

where \( u_0^j = \pi^j(x_0^j) \). The goal of the control problem studied in this work is to solve the following finite time robust optimal control problem,

\[ J_{\pi^j}^{\ast}(x_0^j) = \min_{\pi^j(\cdot)} \{ J_{\pi^j}(x_0^j) \} \]

\[ x_{k+1}^j = Ax_k^j + Bu_k^j + w_k^j \]

\[ u_k^j = \pi^j(x_k^j) \]

\[ x_k^j \in \mathcal{X}, u_k^j \in \mathcal{U}, x_{T^j}, \in \mathcal{O} \]

\[ \forall u_k^j \in W, k \in \{0, \ldots, T^j\} \]

where the time \( T^j \) at which the control task is completed is a design parameter. We present a strategy to iteratively design a feedback policy \( u_k^j = \pi^j(x_k^j) \) with

\[ \pi^j(\cdot) : \mathcal{F}^j \subseteq \mathcal{X} \rightarrow \mathcal{U} \]

solution to Problem (3). In particular the proposed strategy guarantees: i) convergence of the closed-loop system (1) and (4) to a neighborhood of the origin \( \mathcal{O} \), ii) safety, state and input constraints are robustly satisfied, iii) performance improvement, if the controller performs the same task repeatedly (i.e. \( x_0^j = x_0^{j+1} \)), then the worst-case iteration cost (2) is non-decreasing (i.e. \( J_{\pi^j}^{j+1}(x_0^{j+1}) \leq J_{\pi^j}^{j}(x_0^j) \)), and iv) exploration, the domain of the policy (4) is not shrinking with the iteration index (i.e. \( \mathcal{F}^i \subseteq \mathcal{F}^j, \forall j \geq i \)).

Throughout this paper we use the standard function classes \( K, K_\infty \) and \( KL \) notation (see [24]) and we define the distance from a point \( x \in \mathbb{R}^n \) to a set \( \mathcal{O} \subseteq \mathbb{R}^n \) as

\[ |x|_O = \inf_{d \in \mathcal{O}} ||x - d||_1. \]

Furthermore, we make the following assumptions.

Assumption 1: The set \( \mathcal{O} \subseteq \mathbb{R}^n \) is a robust positive invariant set for the autonomous system \( x_{k+1} = (A + BK)x_k + w_k \) and \( w_k \in W \).

\[ \forall x \in \mathcal{O} \rightarrow (A + BK)x_k + w_k \in \mathcal{O}, \forall w_k \in W. \]

Assumption 2: The continuous stage cost \( h(\cdot, \cdot) \) is jointly convex in its arguments. Furthermore, we assume that \( \forall x \in \mathbb{R}^n, \forall u \in \mathbb{R}^d \)

\[ \alpha_x^i(|x|_O) \leq h(x, 0) \leq \alpha_x^u(|x|_O) \]

\[ \alpha_u^i(|u|_{K\infty}) \leq h(0, u) \leq \alpha_u^u(|u|_{K\infty}) \]

where \( \alpha_x^i, \alpha_x^u, \alpha_u^i \) and \( \alpha_u^u \in K_\infty. \)

III. LEARNING MODEL PREDICTIVE CONTROLLER

In this section we illustrate the control design strategy. First, we show how to construct a safe set of states, from which the control policy \( \pi^j(\cdot) \) can successfully complete the control task. Afterward, we define a value function which approximates the cost-to-go associated with the control policy \( \pi^j(\cdot) \). Finally, we exploit the safe set and the value function to design the control policy at the next iteration \( \pi^{j+1}(\cdot) \).

A. Safe Set

In this section we show how to iteratively construct a set of states from which the control task can be safely executed, using the control policy in (4). First, we recall the definition of robust reachable set [25] for the closed-loop system (1) and (4).

\[ R_{k+1}(x_0^j) = \left\{ x_{k+1} \in \mathcal{X} \middle| \exists u_k \in W, x_k \in R_k(x_0^j), x_{k+1} = Ax_k + B \pi^j(x_k) + u_k \right\} \]

with \( R_0(x_0^j) = x_0^j \). The above robust reachable set \( R_k(x_0^j) \) collects that states which may be reached in \( N \)-steps by the closed-loop system (1) and (4).

Now, we define the safe set at the \( j \)th iteration as

\[ SS^j = \bigcup_{k=0}^{T^j} R_k(x_0^j) \bigcup \mathcal{O}. \]

Notice that the above safe set \( SS^j \) contains the state evolution of the closed-loop system (1) and (4) from the initial state \( x_0^j \). In general, robust reachable sets \( R_k(x_0^j) \) are computed propagating the vertices of the disturbance through the system dynamics. Therefore, the computational complexity of constructing the safe set \( SS^j \) exploding with the length of the control task \( T^j \). For this reason in Section IV-A we show how the safe set \( SS^j \) can be approximated using historical data from the closed-loop system (1) and (4).

Finally, we define the convex safe set \( CS^j \) as the convex hull of the union of the safe sets \( SS^k \) for iterations \( k \in \{0, \ldots, j\} \).

\[ CS^j = \text{conv} \left( \bigcup_{k=0}^{j} SS^k \right). \]

Notice that, if the control policies \( \pi^k(\cdot) \) for \( k \in \{0, \ldots, j\} \) safely steer the system to the neighborhood of the origin \( \mathcal{O} \). Then, \( CS^j \) is a robust control invariant set as stated by the following proposition.

Proposition 1: For \( j \geq 1 \), let \( \pi^j(\cdot) : \mathcal{F}^j \rightarrow \mathcal{U} \) be a control policy defined over \( \mathcal{F}^j \subseteq \mathcal{X} \). Consider system (1) in closed-loop with \( \pi^j(\cdot) \) and assume that \( \forall x_0^j \in \mathcal{F}^j \) we have \( x_0^j \in \mathcal{X} \).
and \( x^j_{T_j} \in \mathcal{O} \forall w_k \in \mathcal{W}, k \geq 0 \). Then, the convex safe set \( \mathcal{C} \mathcal{S}^j \) is a robust control invariant set for system (1), i.e.

\[
\forall x \in \mathcal{C} \mathcal{S}^j \rightarrow Ax + B\pi^j(x) + w \in \mathcal{C} \mathcal{S}^j, \forall w \in \mathcal{W}
\]

Proof: By the assumptions on \( \pi^k(\cdot) \) for \( k \in \{0, \ldots, j\} \) and definition (6), we have that \( \mathcal{S} \mathcal{S}^k \) is a robust control invariant set for \( k \in \{0, \ldots, j\} \). Therefore, by linearity of system (1) it follows that \( \mathcal{C} \mathcal{S}^j \) is a robust control invariant set.

\[\]\\

**B. Q-function**

In this section we define the value function \( Q^j(\cdot) : \mathcal{C} \mathcal{S}^j \rightarrow \mathbb{R} \), which approximates the cost-to-go from any state \( x \in \mathcal{C} \mathcal{S}^j \). First we recall the Bellman recursion (2) for the control policy \( \pi^j(\cdot) \)

\[
J^j_{\pi^j}(x) = \max_{w \in \mathcal{W}} \left[ h(x, \pi^j(x)) + J^j_{\pi^j}(Ax + B\pi^j(x) + w) \right], \quad (8)
\]

which represents the worst-case cost-to-go from any point in the state space. The solution to the above Bellman recursion is hard to compute (14) and closed-form exists just for few problems (25). For a survey on strategies to approximate the Bellman recursion we refer to (14), (15).

Now, we defined the worst-case cost-to-go over the safe set as

\[
L^j_{\pi^j}(x) = \begin{cases} \max_{w \in \mathcal{W}} [h(x, \pi^j(x)) + L^j_{\pi^j}(x^j_+(w))] & \text{if } x \in \mathcal{S} \mathcal{S}^j \\ +\infty & \text{if } x \notin \mathcal{S} \mathcal{S}^j \end{cases} \quad (9)
\]

where \( x_+^j(w) = Ax + B\pi^j(x) + w \). Notice that, for all \( x \in \mathcal{S} \mathcal{S}^j \), the above function coincides with the Bellman equation (6). The difference between \( J^j_{\pi^j}(\cdot) \) and \( L^j_{\pi^j}(\cdot) \) is that the domain of the latter is the safe set \( \mathcal{S} \mathcal{S}^j \) from (6). This fact allows us to use a simple data-based strategy to approximate \( L^j_{\pi^j}(\cdot) \), as shown in Section IV-B.

Finally, for all \( x \in \mathcal{C} \mathcal{S}^j \) we define the function

\[
Q^j(x) = \min_{\mu} \{ h(x, \mu(x)) \in \mathcal{W} \} \left\{ \max_{k=0}^N \mu(x) \right\}, \quad (10)
\]

which interpolates the worst-case cost-to-go functions \( L_{\pi^j_k}(\cdot) \) for \( k \in \{0, \ldots, j\} \). Notice that the above \( Q^j(\cdot) \) is simply a convexification of the cost-to-go functions (i.e. \( \text{epi}(Q^j(x)) = \text{conv}(\cup_{k=0}^N \text{epi}(L_{\pi^j_k}(x^k))) \)). Furthermore, if the control policies \( \pi^k(\cdot) \) for \( k \in \{0, \ldots, j\} \) are uniformly steered by the neighborhood of the origin \( \mathcal{O} \), then the approximated value function \( Q^j(\cdot) \) is a robust control Lyapunov function over the convex safe set \( \mathcal{C} \mathcal{S}^j \) for system (1), as shown by the following proposition.

**Proposition 2:** For \( j \geq 1 \), let \( \pi^j(\cdot) : \mathcal{F} \rightarrow \mathcal{U} \) be a control policy defined over \( \mathcal{F} \subseteq \mathcal{X} \). Consider system (1) in closed-loop with \( \pi^j(\cdot) \) and assume that \( \forall x_0^j \in \mathcal{F} \) we have \( x_{T_j}^j \in \mathcal{X} \) and \( x_{T_j}^j \in \mathcal{O} \forall w_k \in \mathcal{W}, k \geq 0 \). Then, \( Q^j(\cdot) \) is a robust control Lyapunov function, i.e.

\[
\min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \left( Q^j(Ax + Bu + w) + h(x, u) - Q^j(x) \right) \leq 0
\]

for all \( x \in \mathcal{C} \mathcal{S}^j \).

\[\]

**Proof:** By the assumptions on \( \pi^k(\cdot) \) for \( k \in \{0, \ldots, j\} \) and definitions (9) and (10), we have that \( \forall x^k \in \mathcal{S} \mathcal{S}^k, k \in \{0, \ldots, j\} \)

\[
\max_{w \in \mathcal{W}} Q^j(Ax^k + Bu^k + w) + h(x, u^k) - Q^j(x^k) \leq 0 \quad (11)
\]

for \( u^k = \pi^k(x^k) \). From (10) we have that if \( x \in \mathcal{C} \mathcal{S}^j \), then we can find some multipliers \( \lambda^k \geq 0 \) for \( k \in \{0, \ldots, j\} \) such that \( \sum_{k=0}^j \lambda^k = 1, \sum_{k=0}^j \lambda^k x^k = x \) and \( \sum_{k=0}^j \lambda^k Q^j(x^k) = Q^j(x) \). Now, we notice that by the Assumption 2 and (11) we have that \( \forall x \in \mathcal{C} \mathcal{S}^j \)

\[
Q^j(x) = \sum_{k=0}^j \lambda^k Q^j(x^k)
\]

\[
\geq \sum_{k=0}^j \lambda^k \max_{w \in \mathcal{W}} \left[ Q^j(Ax^k + Bu^k + w) + h(x, u^k) \right]
\]

\[
\geq \max_{w \in \mathcal{W}} Q^j(Ax + Bu + w) + h(x, u)
\]

for \( u = \sum_{k=0}^j \lambda^k u^k = \sum_{k=0}^j \lambda^k \pi^k(x^k) \in \mathcal{U} \).

**C. Controller Design**

In this section we illustrate the controller’s design which exploits the convex safe set (7) and the approximated value function (10). At each time \( t \) of the \( j \)-th iteration, we design and solve the following finite time optimal control problem

\[
J^j_{\text{finite}}(x^j_t) = \min_{\pi^j_t} \{ \sum_{k=0}^{t-N} h(x^j_{k,t}, u^j_k) + Q^{j-1}(x^j_{t+N}) \}
\]

\[
\begin{align*}
x^j_{k+1,t} & = Ax^j_{k,t} + Bu^j_k + w^j_k \quad k \in \{0, \ldots, t-N\} \\
u^j_k & \in \mathcal{U} \\
x^j_{t+N} & \in \mathcal{C} \mathcal{S}^{j-1} \\
\forall w^j_k \in \mathcal{W} \end{align*}
\]

(12)

where the control policy \( \pi^j_t(\cdot) = [\pi^j_{0,t}(\cdot), \ldots, \pi^j_{t-N,t}(\cdot)] \) and the disturbance \( w^j_t = [w^j_0(\cdot), \ldots, w^j_{t-N}(\cdot)] \). The optimal feedback policy from the above finite time optimal control problem safely steers system (1) from \( x^j_t \) to the convex safe set, while minimizing the worst-case cost.

Let

\[
\pi^j_{t-N,t} = [\pi^j_{0,t}(\cdot), \ldots, \pi^j_{t-N,t}(\cdot)]
\]

(13)

be the optimal feedback policy to Problem (12). Then we apply to system (1)

\[
\pi^j_t(x^j_t) = \pi^j_{t-N,t}(x^j_t)
\]

(14)

The finite time optimal control problem (12) is solved at time \( t+1 \), based on the new state \( x^j_{t+1} \), yielding a moving or receding horizon control strategy.

Furthermore, we define the domain of the LMPC policy (14), which is given by

\[
\mathcal{F}^j = \left\{ x_0 \in \mathcal{X} \left| \exists \kappa(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^d, x_k \in \mathcal{X}, \kappa(x_k) \in \mathcal{U}, x_{k+1} = Ax_k + B\kappa(x_k) + w_k, x_N \in \mathcal{C} \mathcal{S}^{j-1}, \forall w_k \in \mathcal{W}, k \in \{0, \ldots, N\} \right.\right\}
\]

(15)
The set \( F^j \), which collects the feasible initial conditions for Problem (12), is used to compute the initial state \( x^j_0 \) of the \( j \)th iteration. In particular, the initial condition at the \( j \)th iteration is computed solving the following convex optimization problem,

\[
x^j_0 = \arg \max_{x \in F^j} \{ax | a^+x = 0 \}
\]

(16)

where the user-defined row vector \( a \in \mathbb{R}^n \) represents the direction in which the LMPC explores the state space. Finally, \( a^+ \in \mathbb{R}^n \) in (16) is a row vector perpendicular to \( a \).

It is well-known that the solution to Problem (12) can be computed enumerating the vertices of the disturbance over the prediction horizon [26]. Therefore, the computational complexity of Problem (12) explodes with the horizon length \( N \). For this reason, it is important to construct a terminal set and terminal cost, which allow to guarantee safety and performance improvement independently on the prediction horizon length. In the result section, we show that the proposed controller is able to safely explore the state space and to improve its performance, even with a short prediction horizon.

D. Properties

As discussed in Propositions 12 for every point in \( CS^j \) there exists a control policy which safely steers the system to the terminal goal set. The properties of \( CS^j \) and \( Q^j(\cdot) \) are to be used to show that the proposed strategy meets the requirements from Section II. First, we show that the LMPC (12) and (14) safely steers the system to \( \mathcal{O} \).

Theorem 1: Consider system (1) in closed-loop with the LMPC (12) and (14). Let Assumptions 1-2 hold, and initialize \( CS^0 = \mathcal{O} \) and \( Q^0(\cdot) = 0 \). If the initial condition of two subsequent iteration are equal, \( x^j_0 = x^j_{0} \in F^j \). Then, the worst-case iteration cost (2) is non-decreasing with the iteration index \( J^j_{0\rightarrow T+1}(x^j_0) \leq J^j_{0\rightarrow T}(x^j_0) \).

Proof: By Theorem 1 the LMPC (12) and (14) is feasible at time \( t \) of the \( j \)th iteration. Let (13) be the optimal policy time of the \( j \)th iteration and consider the optimal cost,

\[
J^\text{LMPC},j_{t\rightarrow t+N}(x^j_t) = \sum_{k=t}^{t+N-1} h(x^j_{k\mid k}^*, u^j_{k\mid k}^*) + Q^{j-1}(x^j_{t+N\mid t})
\]

\[
\geq h(x^j_{t\mid t}^*, u^j_{t\mid t}^*) + \sum_{k=t}^{t+N-1} h(x^j_{k\mid k}^*, u^j_{k\mid k}^*)
\]

\[
+ \min_{u \in U} \max_{w \in W} \left[ \frac{t+N-1}{N} \left(Ax^j_{t\mid t} + Bu + w \right) + h(x^j_{t\mid t}^*, u) \right]
\]

\[
\geq h(x^j_{t\mid t}^*, u^j_{t\mid t}^*) + \min_{\pi_t(\cdot)} \left[ \sum_{k=t}^{t+N-1} h(x^j_{k\mid k}^*, u^j_{k\mid k}) \right]
\]

\[
= h(x^j_{t\mid t}^*, u^j_{t\mid t}^*) + J^\text{LMPC},j_{t\rightarrow t+N}(x^j_{t})
\]

The above equation and the convergence of the closed-loop system (1) and (14) from Theorem 1 imply that

\[
J^\text{LMPC},j_{0\rightarrow N}(x^j_0) \geq J^j_{0\rightarrow N}(x^j_0) + J^\text{LMPC},j_{1\rightarrow 1+N}(x^j_{1})
\]

\[
\geq \sum_{t=0}^{\infty} h(x^j_{t\mid t}^*, u^j_{t\mid t}^*) + \lim_{t \rightarrow \infty} J^\text{LMPC},j_{t\rightarrow t+N}(x^j_{t})
\]

\[
= \sum_{t=0}^{\infty} h(x^j_{t}, u^j_{t})
\]

The above derivation holds for all disturbance realization, therefore we have that

\[
J^\text{LMPC},j_{0\rightarrow N}(x^j_0) \geq J^j_{0\rightarrow N}(x^j_0)
\]

Finally, we notice that the above inequality together with Equations 9, 10 and the feasibility of the LMPC policy \( \pi_j(\cdot) \) (14) at the next iteration \( j + 1 \) imply that

\[
J^\pi_{j}(x^j_0) = L^j_{\pi_j}(x^j_0)
\]

\[
= \max_{u^{j}_{0},...,u^{j}_{T-1}} \sum_{k=0}^{T-1} h(x^j_{k}, \pi_j(x^j_{k}) + L^j_{\pi}(x^j_{k}))
\]

\[
\geq \max_{u^{j}_{0},...,u^{j}_{T-1}} \sum_{k=0}^{T-1} h(x^j_{k}, \pi_j(x^j_{k}) + Q^j(x^j_{k}))
\]

\[
\geq J^\text{LMPC},j+1_{0\rightarrow N}(x^j_0) \geq J^j_{\pi_{j+1}}(x^j_0) \geq J^j_{\pi_{j+1}}(x^j_0)
\]

Finally, we show that the domain of which the LMPC (12) and (14) does not shrink at each iteration.

Theorem 3: Consider system (1) in closed-loop with the LMPC (12) and (14). Let Assumptions 1-2 hold, and initialize \( CS^0 = \mathcal{O} \) and \( Q^0(\cdot) = 0 \). If \( x^j_0 \in F^j \), \( \forall j \geq 1 \). Then, the
domain of which the LMPC defined in (15) does not shrink at each iteration, i.e. \( F^i \subseteq F^j, \forall j \geq i \).

Proof: The proof follows from the definition of the convex safe set. Notice that by definition (7) we have that \( CS^i \subseteq CS^j, \forall j \geq i \). Therefore, the terminal set in (15) is enlarged at each iteration and consequently \( F^i \subseteq F^j, \forall j \geq i \).

### IV. Practical Implementation

In this section we show how the data from different disturbance realizations may be used to approximate the worst-case cost-to-go function \( L^ j_{\pi}(\cdot) \) in (9). First, we define the realized cost-to-go associated with the stored state \( x_k^j(w^i) \in \tilde{R}_k(x_0^i) \subseteq SS^j \),

\[
J_{k \rightarrow T^j}(x_k^j(w^i)) = \sum_{t=k}^{T^j} h(x_k^j(w^i), u_k^j(w^i))
\]  

(21)

where \( u_k^j(w^i) = \pi^j(x_k^j(w^i)). \)

The realized cost (21) associated with the realized trajectory \( x_k^j(w^i) \) is used to approximate the worst-case cost-to-go function \( L^ j_{\pi}(\cdot) \), over the approximated robust reachable set \( \tilde{R}_k(x_0^i) \) from (18). First, we compute an hyperplane which upper-bounds the realized cost \( J_{k \rightarrow T^j}(x_k^j(w^i)) \) for all stored states \( \{ \bigcup_{j=1}^{M} x_k^j(w^i) \} \subseteq \tilde{R}_k(x_0^i). \) In particular, for time \( k \) of the \( j \)th iteration we defined the hyperplane \( a_k^jx + b_k^j \), where

\[
[a_k^j, b_k^j] = \arg\min_{a \in \mathbb{R}^n, b \in \mathbb{R}} \sum_{i=0}^{M} ||ax_k^i(w^i) + b - J_{k \rightarrow T^j}(x_k^i(w^i))||_2^2 \\
\text{s.t.} \ ax_k^i(w^i) + b \geq J_{k \rightarrow T^j}(x_k^i(w^i)), \forall i \in \{0, \ldots, M\}.
\]

(22)

Finally at the \( j \)th iteration, the hyperplanes \( a_k^jx + b_k^j \) are used to approximate the worst-case cost-to-go \( L^ j_{\pi}(\cdot) \) from (9) as follows,

\[
\hat{L}^ j_{\pi}(x) = \begin{cases} 
+\infty & \text{If } x \notin SS^j \\
0 & \text{Elseif } x \in S, \\
\min_{k \in K^j} \mathbb{I}_{R_k(x_0)}(a_k^jx + b_k^j) & \text{Else}
\end{cases}
\]

(23)

where the set \( K^j = \{0, \ldots, T^j\} \) and the indicator function \( \mathbb{I}_{S}(x) = 1 \) is defined over the set \( S \). The resulting approximated value function is defined as

\[
\hat{Q}^ j(x) = \min_{\mu} \mu \mid (x, \mu) \in \bigcup_{k=0}^{j} \text{conv}(\text{epi}(\hat{L}^ j_{\pi}(x^j))).
\]

(24)

Finally, we underline that the above approximated value function is not a control Lyapunov function for system (1). Indeed, there is a probability \( \gamma > 0 \) that Equation (11) does not hold and \( \hat{Q}^ j(\cdot) \) is not decreasing along the closed-loop trajectory,

\[
\Pr(\hat{Q}^ j(Ax + B\pi^j(x) + w) + h(x, \pi^j(x)) - \hat{Q}^ j(x) > 0) \geq \gamma.
\]

(25)

In the result section, we show that above probability is inversely proportional to the number \( M \) of realized trajectories used to construct \( \hat{L}^ j_{\pi}(\cdot) \) from (23).
V. RESULTS

We test the proposed control strategy on the following double integrator system

\[
x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + w_k,
\]

where the the random disturbance \( w_k \) is uniformly distributed on the set \( W = \{ w \in \mathbb{R}^2 : ||w_k||_\infty \leq 0.1 \} \). The system is subjected to the following state and input constraints, \( x_k \in \mathcal{X} = \{ x \in \mathbb{R}^2 : ||x||_\infty \leq 10 \} \) and \( u_k \in \mathcal{U} = \{ u \in \mathbb{R}^2 : ||u||_\infty \leq 1 \} \), for all \( k \geq 0 \). Furthermore, we compute the minimal robust positive invariant set \( O \) for the autonomous system \( x_{k+1} = (A + BK)x_k + w_k \) where \( -K \) is the LQR gain for \( Q = 1 \) and \( R = 1 \). Finally, we define the stage cost \( h(x, u) = ||x||_O + ||u||_{KO} \) which satisfies Assumption 2.

The convex safe set \( \tilde{CS}^j \) and value function \( \tilde{Q}^j(\cdot) \), used in the LMPC (12) and (14), are approximated as described in Section IV. In particular at each iteration \( j \), we use \( M \) closed-loop realized trajectories to compute the approximated safe set \( \tilde{CS}^j \) and value function \( \tilde{Q}^j(\cdot) \). In order to initialize the LMPC we set \( N = 3 \), \( \tilde{CS}^0 = \emptyset \) and \( \tilde{Q}^0(\cdot) = 0 \). Finally, at each \( j \)th iteration, the initial state \( x_0^j \) is computed as the furthest point along the negative \( x \)-axis which belongs to \( \mathcal{F}^j \). Basically, we set \( a = [-1, 0] \) in (16).

A. Convex Safe Set and Value Function Approximation

In this section, we construct \( \tilde{CS}^1 \) and \( \tilde{Q}^1(\cdot) \) using \( M = 100 \) and \( M = 1000 \) closed-loop trajectories. Furthermore, we perform 1000 Monte-Carlo simulations for the closed-loop system (11) and (14), in order to estimate the properties of \( \tilde{CS}^1 \) and \( \tilde{Q}^1(\cdot) \). First, we estimate the probability (20) that the closed-loop system evolves outside \( \tilde{CS}^j \), given that \( x \in \tilde{CS}^j \). Afterwards, we estimate the probability (25) that \( \tilde{Q}^1(\cdot) \) is not a decreasing along the closed-loop trajectory.

![Fig. 1](image1.png) The approximated robust reachable sets \( \tilde{R}_k(x_0^j) \) used to construct \( \tilde{CS}^j \) with \( M = 100 \) and \( M = 1000 \) realized trajectories. Notice that the approximated convex safe set \( \tilde{CS}^j \) constructed using 1000 trajectories contains the one constructed using 100.

![Fig. 2](image2.png) Approximated value function \( \tilde{Q}^1(\cdot) \) constructed with \( M = 100 \) and \( M = 1000 \) realized trajectories. Note that as more trajectories are used the value of \( \tilde{Q}^1(\cdot) \) increases almost everywhere, thus it better approximated \( Q^1(\cdot) \).

Finally, we analyze how the number of realized trajectories affects the approximated value function \( \tilde{Q}^1(\cdot) \). Figure 2 shows the approximated value function \( \tilde{Q}^1(\cdot) \) constructed with \( M = 100 \) and \( M = 1000 \) realized trajectories. First, we notice that the domain of approximated value function \( Q^1(\cdot) \) is enlarged as more realized trajectories are used to compute the approximation. Indeed, the domain of \( \tilde{Q}^1(\cdot) \) is the approximated safe set \( \tilde{CS}^1 \) from Figure 1. Second, we recall that \( Q^1(\cdot) \) is constructed based on sampled disturbance sequences and it underestimates \( \tilde{Q}^1(\cdot) \), which considers the whole disturbance support. Therefore, we expect that as more sample disturbance sequences are considered \( \tilde{Q}^1(\cdot) \) better approximates \( Q^1(\cdot) \). This intuition is confirmed by Figure 3 we notice that \( \tilde{Q}^1(\cdot) \) constructed with 1000 trajectories better approximates \( Q^1(\cdot) \), because it upper-bounds almost everywhere the value function \( Q^1(\cdot) \) constructed with 100 trajectories. Finally, we recall from Equation (25) that \( Q^1(\cdot) \) is not a control Lyapunov function. Indeed, there is
a probability $\gamma > 0$ that $\hat{Q}^j(\cdot)$ is not decreasing along the realized trajectory of the closed-loop system. In order to estimate the probability $\gamma$, we use 1000 Monte Carlo simulations. As expected, the probability $\gamma$ decreases as more closed-loop trajectories are used to construct $\hat{Q}_j^j(\cdot)$. In particular, we have $\gamma \sim 10.1\%$ and $\gamma \sim 4.3\%$ for $M = 100$ and $M = 1000$, respectively.

B. Iterative Policy Update

In this Section we run the LMPC for 10 iterations. In particular, at each $j$th iteration we collect $M = 1000$ realized trajectories which are used to compute the approximated convex safe set $\hat{CS}^j$ and the approximated value function $\hat{Q}_j^j(\cdot)$. We show that the LMPC is able to explore the state space while safely steering the system to the terminal set $O$.

| INITIAL CONDITION $x_0^j$ AT EACH $j$TH ITERATION. |
|-------------------------------------------------|
| $x_0^0 = -[2.00 \; 0]^T$ | $x_0^6 = -[9.90 \; 0]^T$ |
| $x_0^1 = -[5.46 \; 0]^T$ | $x_0^7 = -[9.90 \; 0]^T$ |
| $x_0^2 = -[6.86 \; 0]^T$ | $x_0^8 = -[9.90 \; 0]^T$ |
| $x_0^3 = -[9.35 \; 0]^T$ | $x_0^9 = -[9.90 \; 0]^T$ |
| $x_0^4 = -[9.90 \; 0]^T$ | $x_0^{10} = -[9.90 \; 0]^T$ |

Table I, where we report the initial condition $x_0^j$ as a function of the iteration index. Furthermore, in Figure 3 we show 1000 realized trajectories for the 2nd, 4th and 8th iterations. We notice that at each $j$th iteration the LMPC safely operates the system in regions of the state space which are further from the origin, until the closed-loop trajectory is close to saturate the state constraints.

![Fig. 3. For iterations $j \in \{2, 4, 8\}$ and $i \in \{1, \ldots, 1000\}$ disturbance realizations we show the closed-loop trajectories $x_i^j(w_i^j)$ from (17). Furthermore, we report the initial condition $x_0^j$ which is further from the origin at each iteration.](image)

Finally, in Figure 4 we report the approximated value function $\hat{Q}_j^j(\cdot)$ for the 2nd, 4th and 8th iterations. We recall that the domain of $\hat{Q}_j^j(\cdot)$ is the approximated convex safe set $\hat{CS}^j$, which is enlarged at each iteration. Therefore, as more iterations of the control task are executed, $\hat{Q}_j^j(\cdot)$ approximates the value function on a larger region of the state space, as shown in Figure 4.

![Fig. 4. Approximated value function $\hat{Q}_j^j$ at the 2nd, 4th and 8th iteration. Notice that the domain of $\hat{Q}_j^j$ is enlarged at each iteration.](image)

C. Performance Improvement

In this section we empirically validate Theorem 2. First, we design a LMPC which minimizes the stage cost $\hat{h}(x, u) = 0.1|x|_O + |u|_{K_O}$. Afterwards, we run the closed-loop system for 10 iterations starting from the same initial condition, $x_0^j = -[0, \; 9.9] \forall j \in \{0, \ldots, 9\}$. In order to initialize the LMPC, we use a suboptimal controller which robustly steers system (26) to $O$ and we exploit the noisy closed-loop data to initialize the approximated convex safe set and value function.

Figure 5 shows the realized cost $\tilde{J}_{0 \rightarrow T_j}(x_0^j(w_i^j))$ from (21) and the worst-case realized cost

$$\max_{i \in \{0, \ldots, M\}} \tilde{J}_{1 \rightarrow T_j}(x_0^j(w_i^j))$$ (27)

for 10 iterations. First, we notice that the LMPC is able to improve the worst-case realized cost associated with the suboptimal policy used at the 0th iteration. Furthermore, we underline that the controller performs exactly the same task at each iteration ($x_0^j = x_0^j, \forall j, i \geq 0$). Therefore, as predicted by Theorem 2, the worst-case realized cost (27) decreases at each iteration until it converges within a tolerance of 0.7%.
VI. CONCLUSIONS

In this paper, we proposed a sample-based Learning Model Predictive Controller (LMPC) for linear systems subject to bounded additive uncertainty. First, we used the LMPC policy to construct a safe set and the associated value function. Afterwards, we showed that the proposed strategy allows to guarantee safety and worst-case performance improvement. Finally, we proposed a sampled-based strategy to approximate the safe set and associated value function. We demonstrated the effectiveness of the proposed approach on a numerical example. In particular, we showed that the proposed LMPC is able to safely explore the state space while estimating the value function associated with the control task. Future work concentrates on finding probability bounds, which would allow to characterize the properties of the approximated safe set and approximate value function as a function of the sampled trajectories.

VII. ACKNOWLEDGMENT

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] 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.
[2] 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.
[3] T. Keller, 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.
[4] 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.
[5] K. Chua, R. Calandra, R. McAllister, and S. Levine, “Deep reinforcement learning in a handful of trials using probabilistic dynamics models,” in Advances in Neural Information Processing Systems, 2018, pp. 4759–4770.
[6] E. Terzi, L. Fagiano, M. Farina, and R. Scattolini, “Learning-based predictive control for linear systems: a unitary approach,” arXiv preprint arXiv:1810.12584, 2018.
[7] U. Rosolia and F. Borrelli, “Learning how to autonomously race a car: a predictive control approach,” arXiv preprint arXiv:1901.08184, 2019.
[8] K. S. Lee and J. H. Lee, “Model predictive control for nonlinear batch processes with asymptotically perfect tracking,” Computers & Chemical Engineering, vol. 21, pp. S873–S879, 1997.
[9] 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.
[10] 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.
[11] 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.
[12] P. N. Beuchat, J. Warrington, and J. Lygeros, “Point-wise maximum approach to approximate dynamic programming,” in 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE, 2017, pp. 3694–3701.
[13] D. P. Bertsekas, “Approximate policy iteration: A survey and some new methods,” Journal of Control Theory and Applications, vol. 9, no. 3, pp. 310–335, 2011.
[14] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-dynamic programming, Athena Scientific Belmont, MA, 1996, vol. 5.
[15] 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.
[16] B. Recht, “A tour of reinforcement learning: The view from continuous control,” Annual Review of Control, Robotics, and Autonomous Systems, 2018.
[17] 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.
[18] 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.
[19] J. Lygeros, C. Tomlin, and S. Sastry, “Controllers for reachability specifications for hybrid systems,” Automatica, vol. 35, no. 3, pp. 349–370, 1999.
[20] 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.
[21] 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.
[22] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks: a computationally efficient approach for linear systems,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 3142–3147, 2017.
[23] ———, “Learning model predictive control for iterative tasks. a data-driven control framework.” IEEE Transactions on Automatic Control, 2017.
[24] C. M. Kellett, “A compendium of comparison function results,” Mathematics of Control, Signals, and Systems, vol. 26, no. 3, pp. 339–374, 2014.
[25] F. Borrelli, A. Bemporad, and M. Morari, Predictive control for linear and hybrid systems. Cambridge University Press, 2017.
[26] 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.