Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework.

Ugo Rosolia and Francesco Borrelli

***Abstract*— A Learning Model Predictive Controller (LMPC) for iterative tasks is presented. The controller is reference-free and is able to improve its performance by learning from previous iterations. A safe set and a terminal cost function are used in order to guarantee recursive feasibility and non-decreasing performance at each iteration. The paper presents the control design approach, and shows how to recursively construct terminal set and terminal cost from state and input trajectories of previous iterations. Simulation results show the effectiveness of the proposed control logic.**

I. INTRODUCTION

Control systems autonomously performing a repetitive task have been extensively studied in the literature [1], [2], [3], [4], [5], [6]. One task execution is often referred to as "iteration" or "trial". Iterative Learning Control (ILC) is a control strategy that allows learning from previous iterations to improve its closed-loop tracking performance. In ILC, at each iteration, the system starts from the same initial condition and the controller objective is to track a given reference, rejecting periodic disturbances [1], [3]. The main advantage of ILC is that information from previous iterations are incorporated in the problem formulation at the next iteration and are used to improve the system performance. The possibility of combining MPC with ILC has been explored in [7], where the authors proposed a Model-based Predictive Control for Batch processes (MPCB). The BMPC is based on a time-varying MIMO system that has a dynamic memory of past batches tracking error. The effectiveness of this approach has been shown through experimental results on a nonlinear system [7], and in [8] the authors proved that the tracking error of the BMPC converges to zero as the number of iterations increases. In [9] a model-based iterative learning control has been proposed. The authors incorporated the tracking error of the previous iterations in the control law and used an observer to deal with stochastic disturbances and noises. Also in this case, the authors showed that the tracking error asymptotically converges to zero. Another study on Model Predictive Control (MPC) for repetitive tasks has appeared in [2]. The authors successfully achieve zero tracking error using a MPC which uses measurements from previous iterations to modify the cost function. In [10] the authors use the trajectories of previous iterations to linearize the model used in the MPC algorithm. The authors proved zero steady-state tracking error in presence of model mismatch. In the aforementioned papers the control goal is to minimize a tracking error under the presence of disturbances. The reference signal is known in advance and does not change at each iteration.

Ugo Rosolia and Francesco Borrelli are with the Department of Mechanical Engineering, University of California at Berkeley , Berkeley, CA 94701, USA {ugo.rosolia, fborrelli} @ berkeley.edu

In this paper we are interested in repetitive tasks where the reference trajectory it is not known. In general, a reference trajectory that maximize the performance over an infinite horizon may be challenging to compute for a system with complex nonlinear dynamics or with parameter uncertainty. These systems include race and rally cars where the environment and the dynamics are complex and not perfectly known [11], [12], or bipedal locomotion with exoskeletons where the human input is unknown apriori and can change at each iteration [13], [14].

Our objective is to design a reference-free iterative control strategy able to learn from previous iterations. At each iteration the cost associated with the closed-loop trajectory shall not increase and state and input constraints shall be satisfied. Nonlinear Model Predictive control is an appealing technique to tackle this problem for its ability to handle state and inputs constraints while minimizing a finite-time predicted cost [15]. However, the receding horizon nature can lead to infeasibility and it does not guaranty improved performance at each iteration [16].

The contribution of this paper is threefold. First we present a novel reference-free learning MPC design for an iterative control task. At each iteration, the initial condition, the constraints and the objective function do not change. The j-th iteration cost is defined as the objective function evaluated for the j-th closed loop system trajectory. Second, we show how to design a terminal safe set and a terminal cost function in order to guarantee that *(i):* the j-th iteration cost does not increase compared to the j-1-th iteration cost (non-increasing cost at each iteration), *(ii):* state and input constraints are satisfied at iterations j if they were satisfied at iteration j-1 (recursive feasibility), *(iii):* the closed-loop equilibrium is asymptotically stable. Third, we assume that the system converges to a steady state trajectory as the number of iteration j goes to infinity and we prove the local optimality of such trajectory.

This paper is organized as follows: in Section II we formally define an iterative task and its j-th iteration cost. The control strategy is illustrated in Section III. Firstly, we show the recursive feasibility and stability of the control logic and, afterwards, we prove the convergence properties. Finally, in Section IV and V, we test the proposed control logic on an infinite horizon linear quadratic regulator with constraints and on a minimum time Dubins car problem. Section VI and VII provide final remarks.

II. PROBLEM DEFINITION

Consider the discrete time system

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where the dynamic update $f(\cdot,\cdot)$ is continuous, $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the system state and input, respectively, subject to the constraints

$$x_t \in \mathcal{X},\ u_t \in \mathcal{U},\ \forall t \in \mathbb{Z}_{0+}. \tag{2}$$

At the j-th iteration the vectors

$$\mathbf{u}^j = [u_0^j,\ u_1^j,\ ...,\ u_t^j,\ ...], \tag{3a}$$

$$\mathbf{x}^j = [x_0^j,\ x_1^j,\ ...,\ x_t^j,\ ...], \tag{3b}$$

collect the inputs applied to system (1) and the corresponding state evolution. In (3), x_t^j and u_t^j denote the system state and the control input at time t of the j-th iteration. We assume that at each j-th iteration the closed loop trajectories start from the same initial state,

$$x_0^j = x_S,\ \forall j \geq 0. \tag{4}$$

The goal is to design a controller which solves the following infinite horizon optimal control problem at each iteration:

$$J^*_{0\to\infty}(x_S) = \min_{u_0,u_1,\dots} \sum_{k=0}^{\infty} h(x_k, u_k) \tag{5a}$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k),\ \forall k \geq 0 \tag{5b}$$

$$x_0 = x_S, \tag{5c}$$

$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \geq 0 \tag{5d}$$

where equations (5b) and (5c) represent the system dynamics and the initial condition, and (5d) are the state and input constraints. The stage cost, $h(\cdot,\cdot)$, in equation (5a) is continuous and it satisfies

$$h(x_F, 0) = 0 \text{ and } h(x_t^j, u_t^j) \succ 0\ \forall\ x_t^j \in \mathbb{R}^n \setminus \{x_F\},\quad u_t^j \in \mathbb{R}^m \setminus \{0\}, \tag{6}$$

where the final state x_F is assumed to be a feasible equilibrium for the unforced system (1)

$$f(x_F, 0) = x_F. \tag{7}$$

Throughout the paper we assume that a local optimal solution to Problem (5) exists and it is denoted as:

$$\begin{aligned} \mathbf{x}^* &= [x_0^*,\ x_1^*,\ ...,\ x_t^*,\ ...], \\ \mathbf{u}^* &= [u_0^*,\ u_1^*,\ ...,\ u_t^*,\ ...]. \end{aligned} \tag{8}$$

Remark 1: The stage cost, $h(\cdot,\cdot)$, in (6) is strictly positive and zero at x_F. Thus, an optimal solution to (5) converges to the final point x_F, i.e. $\lim_{t\to\infty} x_t^* = x_F$.

Remark 2: In practical applications each iteration has a finite-time duration. It is common in the literature to adopt an infinite time formulation at each iteration for the sake of simplicity. We follow such an approach in this paper. Our choice does not affect the practicality of the proposed method.

Next we introduce the definition of the sampled safe set and of the iteration cost. Both which will be used later to guarantee stability and feasibility of the learning MPC.

A. Sampled Safe Set

Definition 1 (one-step controllable set to the set $\mathcal{S}$): For the system (1) we denote the *one-step controllable set to the set $\mathcal{S}$* as

$$\mathcal{K}_1(\mathcal{S}) = \text{Pre}(\mathcal{S}) \cap \mathcal{X}. \tag{9}$$

where

$$\text{Pre}(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n\ :\ \exists u \in \mathcal{U} \text{ s.t. } f(x, u) \in \mathcal{S}\}. \tag{10}$$

$\mathcal{K}_1(\mathcal{S})$ is the set of states which can be driven into the target set $\mathcal{S}$ in one time step while satisfying input and state constraints. N-step controllable sets are defined by iterating $\mathcal{K}_1(\mathcal{S})$ computations.

Definition 2 (N-Step Controllable Set $\mathcal{K}_N(\mathcal{S})$): For a given target set $\mathcal{S} \subseteq \mathcal{X}$, the N-step controllable set $\mathcal{K}_N(\mathcal{S})$ of the system (1) subject to the constraints (2) is defined recursively as:

$$\mathcal{K}_j(\mathcal{S}) \triangleq \text{Pre}(\mathcal{K}_{j-1}(\mathcal{S})) \cap \mathcal{X},\ \ \mathcal{K}_0(\mathcal{S}) = \mathcal{S},\quad j \in \{1, \dots, N\} \tag{11}$$

From Definition 2, all states x_0 of the system (1) belonging to the N-Step Controllable Set $\mathcal{K}_N(\mathcal{S})$ can be driven, by a suitable control sequence, to the target set $\mathcal{S}$ in N steps, while satisfying input and state constraints.

Definition 3 (Maximal Controllable Set $\mathcal{K}_\infty(\mathcal{O})$): For a given target set $\mathcal{O} \subseteq \mathcal{X}$, the maximal controllable set $\mathcal{K}_\infty(\mathcal{O})$ for system (1) subject to the constraints in (2) is the union of all N-step controllable sets $\mathcal{K}_N(\mathcal{O})$ contained in $\mathcal{X}$ ($N \in \mathbb{N}$). We will use controllable sets $\mathcal{K}_N(\mathcal{O})$ where the target $\mathcal{O}$ is a control invariant set [17]. They are special sets, since in addition to guaranteeing that from $\mathcal{K}_N(\mathcal{O})$ we reach $\mathcal{O}$ in N steps, one can ensure that once it has reached $\mathcal{O}$, the system can stay there at all future time instants. These sets are called control invariant set.
Note that x_F in (7) is a control invariant since it is an equilibrium point.

Definition 4 (N-step (Maximal) Stabilizable Set): For a given control invariant set $\mathcal{O} \subseteq \mathcal{X}$, the N-step (maximal) stabilizable set of the system (1) subject to the constraints (2) is the N-step (maximal) controllable set $\mathcal{K}_N(\mathcal{O})$ ($\mathcal{K}_\infty(\mathcal{O})$).

Since the computation of Pre-set is numerically challenging for nonlinear systems, there is extensive literature on how to obtain an approximation (often conservative) of the Maximal Stabilizable Set [18].

In this paper we exploit the iterative nature of the control design and define the *sampled Safe Set* $\mathcal{SS}^j$ at iteration j as

$$\mathcal{SS}^j = \left\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{\infty} x_t^i \right\}. \tag{12}$$

$\mathcal{SS}^j$ is the collection of all state trajectories at iteration i for $i \in M^j$. M^j in equation (12) is the set of indexes k associated with successful iterations k for $k \leq j$, defined as:

$$M^j = \Big\{ k \in [0, j] : \lim_{t \to \infty} x_t^k = x_F \Big\}. \tag{13}$$

From (13) we have that $M^i \subseteq M^j, \forall i \leq j$, which implies that

$$\mathcal{SS}^i \subseteq \mathcal{SS}^j, \forall i \leq j. \tag{14}$$

Figure 1 shows an example of the sampled safe set phase plot, for a two state system.

Remark 3: Note that $\mathcal{SS}^j$ can be interpreted as a sampled subset of the Maximal Stabilizable Set $\mathcal{K}_\infty(x_F)$ as for every point in the set, there exists a feasible control action which satisfies the state constraints and steers the state towards x_F.

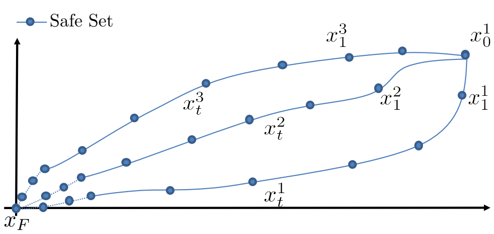

Fig. 1. The circled dots represent the sampled safe set ($\mathcal{SS}$) in a two dimensional phase plane, collecting three successful trajectories.

B. Iteration Cost

At time t of the j-th iteration the cost-to-go associated with the closed loop trajectory (3b) and input sequence (3a) is defined as

$$J_{t \to \infty}^j(x_t^j) = \sum_{k=t}^{\infty} h(x_k^j, u_k^j), \tag{15}$$

where $h(\cdot, \cdot)$ is the stage cost of the problem (5). We define the j-th iteration cost as the cost (15) of the j-th trajectory at time $t = 0$,

$$J_{0 \to \infty}^j(x_0^j) = \sum_{k=0}^{\infty} h(x_k^j, u_k^j). \tag{16}$$

$J_{0 \to \infty}^j(x_0^j)$ quantifies the controller performance at each j-th iteration.

Remark 4: In equations (16)-(15), x_k^j and u_k^j are the realized state and input at the j-th iteration, as defined in (3).

Remark 5: At each j-th iteration the system is initialized at the same starting point $x_0^j = x_S$; thus we have $J_{0 \to \infty}^j(x_0^j) = J_{0 \to \infty}^j(x_S)$.

Finally, we define the function $Q^j(\cdot)$, defined over the sample safe set $\mathcal{SS}^j$ as:

$$Q^j(x) = \begin{cases} \min\limits_{(i,t) \in F^j(x)} J_{t \to \infty}^i(x), & \text{if } x \in \mathcal{SS}^j \\ +\infty, & \text{if } x \notin \mathcal{SS}^j \end{cases}, \tag{17}$$

where $F^j(\cdot)$ in (17) is defined as

$$F^j(x) = \Big\{ (i,t) : i \in [0, j], \ t \geq 0 \text{ with } x = x_t^i; \\ \text{for } x_t^i \in \mathcal{SS}^j \Big\}. \tag{18}$$

Remark 6: The function $Q^j(\cdot)$ in (17) assigns to every point in the sampled safe set, SS^j, the minimum cost-to-go along the trajectories in $\mathcal{SS}^j$ i.e.,

$$\forall x \in SS^j, Q^j(x) = J_{t^* \to \infty}^{i^*}(x) = \sum_{k=t^*}^{\infty} h(x_k^{i^*}, u_k^{i^*}), \tag{19}$$

where the indices pair (i^*, t^*) is the minimizer in (17):

$$(i^*, t^*) = \operatorname*{argmin}_{(i,t) \in F^j(x)} J_{t \to \infty}^i(x), \quad \text{for } x \in \mathcal{SS}^j. \tag{20}$$

In the next section we exploit the fact that at each iteration we solve the same problem to design a controller that guarantees a non-increasing iteration cost (i.e. $J_{0 \to \infty}^j(\cdot) \leq J_{0 \to \infty}^{j-1}(\cdot)$) and which converges to a local optimal solution of (5) (i.e. $\lim_{j \to \infty} \mathbf{x}^j = \mathbf{x}^*$ and $\lim_{j \to \infty} \mathbf{u}^j = \mathbf{u}^*$).

III. LMPC CONTROL DESIGN

In this section we present the design of the proposed Learning Model Predictive Control (LMPC). We first assume that there exists an iteration where the LMPC is feasible at all time instants. Then we prove that the proposed LMPC is guaranteed to be recursively feasible, i.e. feasible at all time instants of every successive iteration. Moreover, the trajectories from previous iterations are used to guarantee non-increasing iterations cost between two successive iterations. Finally, we show that the proposed approach converges to a local optimum of the infinite horizon control problem (5).

A. LMPC Formulation

The LMPC tries to compute a solution to the infinite time optimal control problem (5) by solving at time t of iteration j the finite time constrained optimal control problem

$$J_{t \to t+N}^{\text{LMPC},j}(x_t^j) = \min_{u_{t|t}, \ldots, u_{t+N-1|t}} \Bigg[\sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + \\ + Q^{j-1}(x_{t+N|t}) \Bigg] \tag{21a}$$

s.t.

$$x_{k+1|t} = f(x_{k|t}, u_{k|t}), \ \forall k \in [t, \cdots, t+N-1] \tag{21b}$$

$$x_{t|t} = x_t^j, \tag{21c}$$

$$x_{k|t} \in \mathcal{X}, \ u_k \in \mathcal{U}, \ \forall k \in [t, \cdots, t+N-1] \tag{21d}$$

$$x_{t+N|t} \in \ \mathcal{SS}^{j-1}, \tag{21e}$$

where (21b) and (21c) represent the system dynamics and initial condition, respectively. The state and input constraints are given by (21d). Finally (21e) forces the terminal state into the set $\mathcal{SS}^{j-1}$ defined in equation (12).

Let

$$\begin{aligned}\mathbf{u}^{*,j}_{t:t+N|t} &= [u^{*,j}_{t|t}, \cdots, u^{*,j}_{t+N-1|t}] \\ \mathbf{x}^{*,j}_{t:t+N|t} &= [x^{*,j}_{t|t}, \cdots, x^{*,j}_{t+N|t}]\end{aligned} \tag{22}$$

be the optimal solution of (21) at time t of the j-th iteration and $J^{\text{LMPC},j}_{t\to t+N}(x^j_t)$ the corresponding optimal cost. Then, at time t of the iteration j, the first element of $\mathbf{u}^{*,j}_{t:t+N|t}$ is applied to the system (1)

$$u^j_t = u^{*,j}_{t|t}. \tag{23}$$

The finite time optimal control problem (21) is repeated at time $t+1$, based on the new state $x_{t+1|t+1} = x^j_{t+1}$ (21c), yielding a *moving* or *receding horizon* control strategy.

Assumption 1: At iteration $j = 1$ we assume that $\mathcal{SS}^{j-1} = \mathcal{SS}^0$ is a non-empty set and that the trajectory $\mathbf{x}^0 \in \mathcal{SS}^0$ is feasible and convergent to x_F.

Assumption 1 is not restrictive in practice for a number of applications. For instance, with race cars one can always run a path following controller at very low speed to obtain a feasible state and input sequence.

In the next section we prove that, under Assumption 1, the LMPC (21) and (23) in closed loop with system (1) guarantees recursively feasibility and stability, and non-increase of the iteration cost at each iteration.

Remark 7: From (12), $\mathcal{SS}^j$ at the j-th iteration is the set of all successful trajectories performed in the first j trials. We assume that these trajectories can be recorded and stored at each iteration. Checking if a state is in $\mathcal{SS}^j$ is a simple search. However, the optimization problem (21) becomes challenging to solve even in the linear case due to the integer nature of the constraints (21e). In Section VI.A we comment on practical approaches to improve the computational time to solve (21).

B. Recursive feasibility and stability

As mentioned in Section II, for every point in the set $\mathcal{SS}^j$ there exists a control sequence that can drive the system to the terminal point x_F. The properties of $\mathcal{SS}^j$ and $Q^j(\cdot)$ are used in the next proof to show recursive feasibility and asymptotic stability of the equilibrium point x_F.

Theorem 1: Consider system (1) controlled by the LMPC controller (21) and (23). Let $\mathcal{SS}^j$ be the sampled safe set at iteration j as defined in (12). Let assumption 1 hold, then the LMPC (21) and (23) is feasible $\forall\ t \in \mathbb{Z}_{0+}$ and iteration $j \geq 1$. Moreover, the equilibrium point x_F is asymptotically stable for the closed loop system (1) and (23) at every iteration $j \geq 1$.

The proof follows from standard MPC arguments.

Proof: By assumption $\mathcal{SS}^0$ is non empty. From (14) we have that $\mathcal{SS}^0 \subseteq \mathcal{SS}^{j-1}\ \forall j \geq 1$, and consequently $\mathcal{SS}^{j-1}$ is a non empty set. In particular, there exists a trajectory $\mathbf{x}^0 \in \mathcal{SS}^0 \subseteq \mathcal{SS}^{j-1}$. From (4) we know that $x^j_0 = x_S\ \forall j \geq 0$. At time $t = 0$ of the j-th iteration the N steps trajectory

$$[x^0_0,\ x^0_1,\ ..., x^0_N] \in \mathcal{SS}^{j-1}, \tag{24}$$

and the related input sequence,

$$[u^0_0,\ u^0_1,\ ..., u^0_{N-1}], \tag{25}$$

satisfy input and state constrains (21b)-(21c)-(21d). Therefore (24)-(25) is a feasible solution to the LMPC (21) and (23) at $t = 0$ of the j-th iteration.

Assume that at time t of the j-th iteration the LMPC (21) and (23) is feasible and let $\mathbf{x}^{*,j}_{t:t+N|t}$ and $\mathbf{u}^{*,j}_{t:t+N|t}$ be the optimal trajectory and input sequence, as defined in (22). From (21c) and (23) the realized state and input at time t of the j-th iteration are given by

$$\begin{aligned}x^j_t &= x^{*,j}_{t|t}, \\ u^j_t &= u^{*,j}_{t|t}.\end{aligned} \tag{26}$$

Moreover, the terminal constraint (21e) enforces $x^{*,j}_{t+N|t} \in \mathcal{SS}^{j-1}$ and, from (19),

$$Q^{j-1}(x^{*,j}_{t+N|t}) = J^{i^*}_{t^*\to\infty}(x^{*,j}_{t+N|t}) = \sum_{k=t^*}^{\infty} h(x^{i^*}_k, u^{i^*}_k). \tag{27}$$

Note that $x^{i^*}_{t^*+1} = f(x^{i^*}_{t^*}, u^{i^*}_{t^*})$ and, by the definition of $Q^j(\cdot)$ and $F^j(x)$ in (17)-(18), $x^{i^*}_{t^*} = x^{*,j}_{t+N|t}$. Since the state update in (1) and (21b) are assumed identical we have that

$$x^j_{t+1} = x^{*,j}_{t+1|t}. \tag{28}$$

At time $t+1$ of the j-th iteration the input sequence

$$[u^{*,j}_{t+1|t},\ u^{*,j}_{t+2|t},\ ...,\ u^{*,j}_{t+N-1|t},\ u^{i^*}_{t^*}], \tag{29}$$

and the related feasible state trajectory

$$[x^{*,j}_{t+1|t},\ x^{*,j}_{t+2|t},\ ...,\ x^{*,j}_{t+N-1|t},\ x^{i^*}_{t^*},\ x^{i^*}_{t^*+1}] \tag{30}$$

satisfy input and state constrains (21b)-(21c)-(21d). Therefore, (29)-(30) is a feasible solution for the LMPC (21) and (23) at time $t+1$.

We showed that at the j-th iteration, $\forall j \geq 1$, *(i):* the LMPC is feasible at time $t = 0$ and *(ii):* if the LMPC is feasible at time t, then the LMPC is feasible at time $t+1$. Thus, we conclude by induction that the LMPC in (21) and (23) is feasible $\forall j \geq 1$ and $t \in \mathbb{Z}_{0+}$.

Next we use the fact the Problem (21) is time-invariant at each iteration j and we replace $J^{\text{LMPC},j}_{t\rightarrow t+N}(\cdot)$ with $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$. In order to show the asymptotic stability of x_F we have to show that the optimal cost, $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$, is a Lyapunov function for the equilibrium point x_F (7) of the closed loop system (1) and (23) [17]. Continuity of $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$ can be shown as in [16]. Moreover from (5a), $J^{\text{LMPC},j}_{0\rightarrow N}(x) \succ 0 \ \forall \ x \in \mathbb{R}^n \setminus \{x_F\}$ and $J^{\text{LMPC},j}_{0\rightarrow N}(x_F) = 0$. Thus, we need to show that $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$ is decreasing along the closed loop trajectory.
From (28) we have $x^{*,j}_{t+1|t} = x^j_{t+1}$, which implies that

$$J^{\text{LMPC},j}_{0\rightarrow N}(x^*_{t+1|t}) = J^{\text{LMPC},j}_{0\rightarrow N}(x^j_{t+1}). \tag{31}$$

Given the optimal input sequence and the related optimal trajectory in (22), the optimal cost is given by

$$\begin{aligned}
J^{\text{LMPC},j}_{0\rightarrow N}(x^j_t) = &\min_{u_{t|t},\dots,u_{t+N-1|t}} \Bigg[\sum_{k=0}^{N-1} h(x_{k|t}, u_{k|t}) + \\
&\qquad + Q^{j-1}(x_{N|t}) \Bigg] = \\
&= h(x^{*,j}_{t|t}, u^{*,j}_{t|t}) + \sum_{k=1}^{N-1} h(x^{*,j}_{t+k|t}, u^{*,j}_{t+k|t}) + Q^{j-1}(x^{*,j}_{t+N|t}) = \\
&= h(x^{*,j}_{t|t}, u^{*,j}_{t|t}) + \sum_{k=1}^{N-1} h(x^{*,j}_{t+k|t}, u^{*,j}_{t+k|t}) + J^{i^*}_{t^*\rightarrow\infty}(x^{*,j}_{t+N|t}) = \\
&= h(x^{*,j}_{t|t}, u^{*,j}_{t|t}) + \sum_{k=1}^{N-1} h(x^{*,j}_{t+k|t}, u^{*,j}_{t+k|t}) + \sum_{k=t^*}^{\infty} h(x^{i^*}_k, u^{i^*}_k) = \\
&= h(x^{*,j}_{t|t}, u^{*,j}_{t|t}) + \sum_{k=1}^{N-1} h(x^{*,j}_{t+k|t}, u^{*,j}_{t+k|t}) + h(x^{i^*}_{t^*}, u^{i^*}_{t^*}) + \\
&\qquad + Q^{j-1}(x^{i^*}_{t^*+1}) \geq \\
&\geq h(x^{*,j}_{t|t}, u^{*,j}_{t|t}) + J^{\text{LMPC},j}_{0\rightarrow N}(x^{*,j}_{t+1|t}),
\end{aligned} \tag{32}$$

where i^* is defined in (20).

Finally, from equations (23), (26) and (31)-(32) we conclude that the optimal cost is a decreasing Lyapunov function along the closed loop trajectory,

$$\begin{aligned}
&J^{\text{LMPC},j}_{0\rightarrow N}(x^j_{t+1}) - J^{\text{LMPC},j}_{0\rightarrow N}(x^j_t) \leq -h(x^j_t, u^j_t) < 0, \\
&\qquad \forall \ x^j_t \in R^n \setminus \{x_F\}, \ \forall \ u^j_t \in R^m \setminus \{0\}
\end{aligned} \tag{33}$$

Equation (33), the positive definitiveness of $h(\cdot)$ and the continuity of $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$ imply that x_F is asymptotically stable. ■

C. Convergence properties

In this Section we assume that the LMPC (21) and (23) converges to a steady state trajectory. We show two results. First, the j-th iteration cost $J^j_{0\rightarrow\infty}(\cdot)$ does not worsen as j increases. Second, the steady state trajectory is a local optimal solution of the infinite horizon control problem (5). In this Section we use the fact the Problem (21) is time-invariant at each iteration j and we replace $J^{\text{LMPC},j}_{t\rightarrow t+N}(\cdot)$ with $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$.

Theorem 2: Consider system (1) in closed loop with the LMPC controller (21) and (23). Let $\mathcal{SS}^j$ be the sampled safe set at the j-th iteration as defined in (12). Let assumption 1 hold, then the iteration cost $J^j_{0\rightarrow\infty}(\cdot)$ does not increase with the iteration index j.

Proof: First, we find a lower bound on the j-th iteration cost $J^j_{0\rightarrow\infty}(\cdot)$, $\forall \ j > 0$. Consider the realized state and input sequence (3) at the j-th iteration, which collects the first element of the optimal state and input sequence to the LMPC (21) and (23) at time t, $\forall t \in \mathbb{Z}_{0+}$, as shown in (26). By the definition of the iteration cost in (15), we have

$$\begin{aligned}
J^{j-1}_{0\rightarrow\infty}(x_S) &= \sum_{t=0}^{\infty} h(x^{j-1}_t, u^{j-1}_t) = \\
&= \sum_{t=0}^{N-1} h(x^{j-1}_t, u^{j-1}_t) + \sum_{t=N}^{\infty} h(x^{j-1}_t, u^{j-1}_t) \geq \\
&\geq \sum_{t=0}^{N-1} h(x^{j-1}_t, u^{j-1}_t) + Q^{j-1}(x^{j-1}_N) \geq \\
&\geq \min_{u_0,\dots,u_{N-1}} \left[\sum_{k=0}^{N-1} h(x_k, u_k) + Q^{j-1}(x_N) \right] = \\
&= J^{\text{LMPC},j}_{0\rightarrow N}(x^j_0).
\end{aligned} \tag{34}$$

Then we notice that, at the j-th iteration, the optimal cost of the LMPC (21) and (23) at $t=0$, $J^{\text{LMPC},j}_{0\rightarrow N}(x^j_0)$, can be upper bounded along the realized trajectory (3) using (33)

$$\begin{aligned}
J^{\text{LMPC},j}_{0\rightarrow N}(x^j_0) &\geq h(x^j_0, u^j_0) + J^{\text{LMPC},j}_{0\rightarrow N}(x^j_1) \ \geq \\
&\geq h(x^j_0, u^j_0) + h(x^j_1, u^j_1) + J^{\text{LMPC},j}_{0\rightarrow N}(x^j_2) \ \geq \\
&\geq \lim_{T\rightarrow\infty} \left[\sum_{k=0}^{T-1} h(x^j_k, u^j_k) + J^{\text{LMPC},j}_{0\rightarrow N}(x^j_T) \right].
\end{aligned} \tag{35}$$

Note that $J^{\text{LMPC},j}_{0\rightarrow N}(\cdot)$ evaluated along the j-th closed loop trajectory (3b) is a decreasing function convergent to zero. Furthermore, the sum of the stage costs, $h(\cdot,\cdot)$, in (35) is upper-bounded by the j-th iteration cost at time $t=0$

$J^{\text{LMPC},j}_{0\to N}(x^j_0)$, and therefore the limit in (35) is well defined. Moreover, from Theorem 1 x_F is asymptotically stable for the closed loop system (1) and (23) (i.e. $\lim_{t\to\infty} x^j_t = x_F$), thus

$$\begin{aligned} J^{\text{LMPC},j}_{0\to N}(x^j_0) &\geq \lim_{T\to\infty}\left[\sum_{k=0}^{T-1} h(x^j_k,u^j_k) + J^{\text{LMPC},j}_{0\to N}(x^j_T)\right] = \\ &= \sum_{k=0}^{\infty} h(x^j_k,u^j_k) = J^j_{0\to\infty}(x_S). \end{aligned} \tag{36}$$

From equations (34)-(36) we conclude that

$$J^{j-1}_{0\to\infty}(x_S) \geq J^{\text{LMPC},j}_{0\to N}(x^j_0) \geq J^j_{0\to\infty}(x_S), \tag{37}$$

thus the iteration cost is non-increasing. ∎

Theorem 3: Consider system (1) in closed loop with the LMPC controller (21) and (23). Let $\mathcal{SS}^j$ be the sampled safe set at the j-th iteration as defined in (12). Let assumption 1 hold and assume that the closed loop system (1) and (23) converges to a steady state trajectory $\mathbf{x}^\infty$, for iteration $j \to \infty$. Then, the steady state input $\mathbf{u}^\infty = \lim_{j\to\infty}\mathbf{u}^j$ and the related steady state trajectory $\mathbf{x}^\infty = \lim_{j\to\infty}\mathbf{x}^j$ is a local optimal solution for the infinite horizon optimal control problem (5), i.e., $\mathbf{x}^\infty = \mathbf{x}^*$ and $\mathbf{u}^\infty = \mathbf{u}^*$.

Proof: By assumption, the closed loop system (1) and (23) converges to a steady state trajectory, $\mathbf{x}^\infty$. By definition both the sampled safe set SS^j and the terminal cost function $Q^j(\cdot)$ converge to steady state quantities, denoted as $\mathcal{SS}^\infty$ and $Q^\infty(\cdot)$, respectively. In particular, from definition (12), we have that $\mathbf{x}^\infty \in \mathcal{SS}^\infty$. From (33) we have that

$$\begin{aligned} J^{\text{LMPC},\infty}_{0\to N}(x^\infty_t) &\geq h(x^\infty_t,u^\infty_t) + J^{\text{LMPC},\infty}_{0\to N}(x^\infty_{t+1}) \quad \geq \\ &\geq h(x^\infty_t,u^\infty_t) + h(x^\infty_{t+1},u^\infty_{t+1}) + J^{\text{LMPC},\infty}_{0\to N}(x^\infty_{t+2}) \geq \\ &\geq \lim_{T\to\infty}\left[\sum_{k=0}^{T-1} h(x^\infty_{t+k},u^\infty_{t+k}) + J^{\text{LMPC},\infty}_{0\to N}(x^\infty_{t+T})\right]. \end{aligned} \tag{38}$$

Moreover, from Theorem 1 we have that x_F is asymptotically stable for the closed loop system (1) and (23), thus

$$\begin{aligned} J^{\text{LMPC},\infty}_{0\to N}(x^\infty_t) &\geq \lim_{T\to\infty}\left[\sum_{k=0}^{T} h(x^\infty_{t+k},u^\infty_{t+k}) + J^{\text{LMPC},\infty}_{0\to N}(x^\infty_{t+T})\right] = \\ &= \sum_{k=0}^{\infty} h(x^\infty_{t+k},u^\infty_{t+k}) = \\ &= \sum_{k=0}^{N-1} h(x^\infty_{t+k},u^\infty_{t+k}) + \sum_{k=N}^{\infty} h(x^\infty_{t+k},u^\infty_{t+k}). \end{aligned} \tag{39}$$

Note that in equation (39) $x^\infty_{t+k} \in SS^\infty$, thus

$$\begin{aligned} J^{\text{LMPC},\infty}_{0\to N}(x^\infty_t) &\geq \sum_{k=0}^{N-1} h(x^\infty_{t+k},u^\infty_{t+k}) + \sum_{k=N}^{\infty} h(x^\infty_{t+k},u^\infty_{t+k}) \geq \\ &\geq \sum_{k=0}^{N-1} h(x^\infty_{t+k},u^\infty_{t+k}) + Q^\infty(x^\infty_{t+N}). \end{aligned} \tag{40}$$

In (40) the cost associated with the feasible trajectory

$$\mathbf{x}^\infty_{t:t+N} = [x^\infty_t,\ x^\infty_{t+1},\ \ldots,\ x^\infty_{t+N}] \tag{41}$$

is a lower bound on the optimal cost $J^{\text{LMPC},\infty}_{0\to N}(x^\infty_t)$. Therefore, the trajectory $\mathbf{x}^\infty_{t:t+N}$ and the related input sequence

$$\mathbf{u}^\infty_{t:t+N} = [u^\infty_t,\ u^\infty_{t+1},\ \ldots,\ u^\infty_{t+N-1}] \tag{42}$$

is an optimal solution to the LMPC (21) and (23) at time t of the j-th iteration for $j \to \infty$.
Next, we prove that $\mathbf{x}^\infty_{0:N+1}$ and $\mathbf{u}^\infty_{0:N+1}$ is a locally optimal solution to the LMPC (21) and (23) where N is replaced with $N+1$. Consider the Hamiltonian

$$H(x,u,\lambda) = h(x,u) + f(x,u)^T\lambda, \tag{43}$$

where $h(x,u)$ and $f(x,u)$ are the stage cost and the system dynamics defined in equations (21) and (1), respectively. The minimum principle [19] states that, if the state trajectory is optimal, it exists a sequence of costate λ^∞_k such that:

$$x^\infty_{k+1} = H_\lambda(x^\infty_k, u^\infty_k, \lambda^\infty_{k+1}) \tag{44a}$$
$$\lambda^\infty_k = H_x(x^\infty_k, u^\infty_k, \lambda^\infty_{k+1}) \tag{44b}$$
$$u^\infty_k = \operatorname*{argmin}_{\bar{u}_k} H(x^\infty_k, \bar{u}_k, \lambda^\infty_{k+1}). \tag{44c}$$

Therefore, for the optimal solution to the LMPC (21) and (23) at time $t = 0$ of the j-th iteration for $j \to \infty$, defined in (41)-(42),

$$\begin{aligned} \mathbf{x}^\infty_{0:N} &= [x^\infty_0,\ x^\infty_1,\ \ldots,\ x^\infty_N] \\ \mathbf{u}^\infty_{0:N} &= [u^\infty_0,\ u^\infty_1,\ \ldots,\ u^\infty_{N-1}], \end{aligned} \tag{45}$$

it exists a sequence of costate

$$[\lambda^\infty_1,\ \lambda^\infty_2,\ \ldots,\ \lambda^\infty_N], \tag{46}$$

that satisfies the minimum principle (44). Moreover, for the optimal solution of the LMPC (21) and (23) at time $t = 1$ of the j-th iteration for $j \to \infty$,

$$\begin{aligned} \mathbf{x}^{\infty}_{1:1+N} &= [x^{\infty}_1,\ x^{\infty}_2,\ ...,\ x^{\infty}_{N+1}] \\ \mathbf{u}^{\infty}_{1:1+N} &= [u^{\infty}_1,\ u^{\infty}_2,\ ...,\ u^{\infty}_N], \end{aligned} \tag{47}$$

there exists a vector $\bar{\lambda}^{\infty}$ such that

$$[\bar{\lambda}^{\infty}_2,\ \bar{\lambda}^{\infty}_3,\ ...,\ \bar{\lambda}^{\infty}_{N+1}] \tag{48}$$

satisfies the minimum principle (44). Therefore, from equations (43) and (44c) we have that

$$\begin{aligned} u^{\infty}_k &= h(x^{\infty}_k, u^{\infty}_k) + f(x^{\infty}_k, u^{\infty}_k)^T \lambda^{\infty}_{k+1} \quad \forall k \in [0, N-1] \\ u^{\infty}_k &= h(x^{\infty}_k, u^{\infty}_k) + f(x^{\infty}_k, u^{\infty}_k)^T \bar{\lambda}^{\infty}_{k+1} \quad \forall k \in [1, N], \end{aligned} \tag{49}$$

from this

$$\lambda^{\infty}_{k+1} = \bar{\lambda}^{\infty}_{k+1},\ \forall k \in [1, N-1]. \tag{50}$$

Finally, we conclude that the $N + 1$ steps trajectory

$$\begin{aligned} \mathbf{x}^{\infty}_{0:N+1} &= [x^{\infty}_0,\ x^{\infty}_1,\ ...,\ x^{\infty}_{N+1}] \\ \mathbf{u}^{\infty}_{0:N+1} &= [u^{\infty}_0,\ u^{\infty}_1,\ ...,\ u^{\infty}_N], \end{aligned} \tag{51}$$

and the costate sequence

$$\tilde{\lambda}^{\infty} = [\lambda^{\infty}_1,\ \lambda^{\infty}_2 = \bar{\lambda}^{\infty}_2,\ ...,\ \lambda^{\infty}_N = \bar{\lambda}^{\infty}_N,\ \bar{\lambda}^{\infty}_{N+1}] \tag{52}$$

satisfy the minimum principle. Therefore at time $t = 0$ of the j-th iteration for $j \to \infty$, the trajectory and its related input sequence,

$$\begin{aligned} \mathbf{x}^{\infty}_{0:N+1} &= [x^{\infty}_0,\ x^{\infty}_1,\ ...,\ x^{\infty}_{N+1}] \\ \mathbf{u}^{\infty}_{0:N+1} &= [u^{\infty}_0,\ u^{\infty}_1,\ ...,\ u^{\infty}_N], \end{aligned} \tag{53}$$

is a local optimal solution for the LMPC (21) and (23) with horizon $N+1$ steps. Next, we show that the above procedure can be iterated to prove local optimality of the $N + 2$ steps trajectory $\mathbf{x}^{\infty}_{[0:N+2]}$ and the related input sequence $\mathbf{u}^{\infty}_{[0:N+2]}$ for the LMPC (21) and (23) with $N = N + 2$. Let

$$[\hat{\lambda}^{\infty}_3,\ \hat{\lambda}^{\infty}_4,\ ...,\ \hat{\lambda}^{\infty}_{N+2}] \tag{54}$$

be the costate associated with the solution of the LMPC at time $t = 2$ of the j-th iteration for $j \to \infty$

$$\begin{aligned} \mathbf{x}^{\infty}_{2:2+N} &= [x^{\infty}_2,\ x^{\infty}_3,\ ...,\ x^{\infty}_{N+2}] \\ \mathbf{u}^{\infty}_{2:2+N} &= [u^{\infty}_2,\ u^{\infty}_3,\ ...,\ u^{\infty}_{N+1}]. \end{aligned} \tag{55}$$

We have, from equations (43) and (44c), and optimality of the trajectory in (53), that

$$\begin{aligned} u^{\infty}_k &= h(x^{\infty}_k, u^{\infty}_k) + f(x^{\infty}_k, u^{\infty}_k)^T \tilde{\lambda}^{\infty}_{k+1} \quad \forall k \in [0, N] \\ u^{\infty}_k &= h(x^{\infty}_k, u^{\infty}_k) + f(x^{\infty}_k, u^{\infty}_k)^T \hat{\lambda}^{\infty}_{k+1} \quad \forall k \in [2, N+1], \end{aligned} \tag{56}$$

from this

$$\tilde{\lambda}^{\infty}_{k+1} = \hat{\lambda}^{\infty}_{k+1},\ \forall k \in [2, N]. \tag{57}$$

Therefore the $N + 2$ steps trajectory and the related costate

$$\begin{aligned} \mathbf{x}^{\infty}_{0:N+2} &= [x^{\infty}_0,\ x^{\infty}_1,\ ...,\ x^{\infty}_{N+2}] \\ \mathbf{u}^{\infty}_{0:N+2} &= [u^{\infty}_0,\ u^{\infty}_1,\ ...,\ u^{\infty}_{N+1}], \\ [\tilde{\lambda}^{\infty}_1, \tilde{\lambda}^{\infty}_2,\ \tilde{\lambda}^{\infty}_3 &= \hat{\lambda}^{\infty}_3,\ ...,\ \tilde{\lambda}^{\infty}_{N+1} = \hat{\lambda}^{\infty}_{N+1},\ \hat{\lambda}^{\infty}_{N+2}] \end{aligned} \tag{58}$$

satisfy the minimum principle and it is locally optimal for the LMPC (21) and (23) with horizon $N + 2$ steps. Iterating this procedure we conclude that $\mathbf{x}^{\infty}$ and its related input sequence, $\mathbf{u}^{\infty}$, is a local optimal solution to the LMPC (21) and (23) defined over the infinite horizon and thus is a local optimal solution of the infinite horizon control problem (5),

$$\begin{aligned} \mathbf{x}^{\infty} &= \mathbf{x}^*, \\ \mathbf{u}^{\infty} &= \mathbf{u}^*. \end{aligned} \tag{59}$$

■

Remark 8: Given a locally optimal solution to the LMPC (21) and (23) defined over infinite horizon, $\mathbf{x}^{\infty}$, we have that $\lim_{t\to\infty} x^j_t = x_F$. Therefore, the terminal constraints (21e) is trivially satisfied and the terminal cost, $Q^{j-1}(\cdot)$, vanishes. Thus, every local optimal solution to the LMPC (21) and (23) for $N \to \infty$ is a locally optimal solution for the infinite horizon control problem (5). Obviously, the terminal constraint and terminal cost are necessary to guarantee the properties of the LMPC (21) and (23) proved in Theorems (1)-(3).

IV. Examples

A. Constrained LQR controller

In this section, we test the proposed LMPC on the following infinite horizon linear quadratic regulator with constraints (CLQR)

$$J^*_{0\to\infty}(x_S) = \min_{u_0, u_1, ...} \sum_{k=0}^{\infty} \Big[||x_k||_2^2 + ||u_k||_2^2 \Big] \tag{60a}$$

$$\text{s.t. } x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k,\ \forall k \geq 0 \tag{60b}$$

$$x_0 = [-3.95 \quad -0.05]^T, \tag{60c}$$

$$\begin{bmatrix} -4 \\ -4 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 4 \end{bmatrix} \ \forall k \geq 0 \tag{60d}$$

$$-1 \leq u_k \leq 1 \ \ \forall k \geq 0. \tag{60e}$$

Firstly, we compute a feasible solution to (60) using an open loop controller that drives the system close to the origin and, afterwards, an unconstrained LQR feedback controller. This feasible trajectory is used to construct the sampled safe set, $\mathcal{SS}^0$, and the terminal cost, $Q^0(\cdot)$, needed to initialize the first iteration of the LMPC (21) and (23).

The LMPC (21) and (23) is implemented with the quadratic running cost $h(x_k, u_k) = ||x_k||_2^2 + ||u_k||_2^2$, an horizon length $N = 4$, and the states and input constraints (60d)-(60e). The LMPC (21) and (23) is reformulated as a Mixed Integer Quadratic Programming and it is implemented in YALMIP [20] using the solver bonmin [21]. Each j-th iteration has an unknown fixed-time duration, $\tilde{t}_j$, defined as

$$\tilde{t}_j = \min\Big\{t \in \mathbb{Z}_{0+} : J_{0\to N}^{\text{LMPC},j}(x_t^j) \leq \epsilon\Big\}. \tag{61}$$

with $\epsilon = 10^{-8}$. Furthermore, each j-th closed loop trajectory is used to enlarge the sampled safe set used at the j+1-th iteration.

After 9 iterations, the LMPC converges to steady state solution $\mathbf{x}^\infty = \mathbf{x^9}$ with a tollerance of γ:

$$\max_{t\in[0,\tilde{t}_9]} ||x_t^9 - x_t^8||_2 < \gamma \tag{62}$$

with $\gamma = 10^{-10}$. Table I reports the number of points in the sampled safe set at each j-th iteration, until convergence is reached.

TABLE I
NUMBER OF POINTS IN THE SAMPLED SAFE SET.

Iteration	Number of Points
$j = 1$	62
$j = 2$	77
$j = 3$	92
$j = 4$	107
$j = 5$	122
$j = 6$	137
$j = 7$	152
$j = 8$	167
$j = 9$	182

We observe that the iteration cost is non-increasing over the iterations and the LMPC (21) and (23) improves the closed loop performance, as shown in Table II.

We compare this steady state trajectory with the exact solution of the CLQR (60), which is computed using the algorithm in [17]. In Table III is reported the deviation error,

$$\sigma_t = ||x_t^\infty - x_t^*||_2, \tag{63}$$

which quantifies, at each time step t, the distance between the optimal trajectory $\mathbf{x}^*$ of the CLQR (60) and steady state trajectory $\mathbf{x}^\infty$ at which the LMPC (21) and (23) has converged. We notice that the maximum deviation error is $\max[\sigma_0, \dots, \sigma_{\tilde{t}_\infty}] = 1.62 \times 10^{-5}$, and that the 2-norm of the difference between the exact optimal cost and the cost

TABLE II
COST OF THE LMCPC AT EACH j-TH ITERATION

Iteration	Iteration Cost
$j = 0$	57.1959612323
$j = 1$	49.9313760802
$j = 2$	49.9166093038
$j = 3$	49.9163668249
$j = 4$	49.9163602456
$j = 5$	49.9163600500
$j = 6$	49.9163600443
$j = 7$	49.9163600441
$j = 8$	49.9163600440
$j = 9$	49.9163600440

associated with the steady state trajectory is $||J_{0\to\infty}^*(x_0^*) - J_{0\to\infty}^\infty(x_0^\infty)||_2 = 1.565 \times 10^{-20}$. Therefore, we confirm that the LMPC (21) and (23) has converged to a locally optimal solution that in the specific case is the global optimal solution.

Finally, Figures 2-3 show the steady state trajectory and

TABLE III
DEVIATION ERROR.

Time Step t	Error
$t = 1$	$1.299 \cdot 10^{-14}$
$t = 2$	$8.516 \cdot 10^{-8}$
$t = 3$	$1.008 \cdot 10^{-8}$
$t = 4$	$1.392 \cdot 10^{-7}$
$t = 5$	$1.473 \cdot 10^{-7}$
$t = 6$	$1.231 \cdot 10^{-7}$
$t = 7$	$8.395 \cdot 10^{-8}$
$t = 8$	$1.376 \cdot 10^{-7}$
$t = 9$	$4.180 \cdot 10^{-7}$
$t = 10$	$1.043 \cdot 10^{-6}$
$t = 11$	$2.140 \cdot 10^{-6}$
$t = 12$	$3.494 \cdot 10^{-6}$
$t = 13$	$4.996 \cdot 10^{-6}$
$t = 14$	$1.641 \cdot 10^{-6}$

the related input sequence. It is interesting to notice that the steady state solution to the LMPC (21) and (23) saturates both state and input constraints as the exact solution to the CLQR (60).

B. Dubins Car with Obstacle and Acceleration Saturation

In this section, we test the proposed LMPC on the minimum time Dubins car problem [22] in discrete time. In this example, we add a known saturation limit on the acceleration in order to simulate the behavior of the friction circle [23], [24]. We control the car acceleration and steering. The controller's goal is to steer the system from the starting point x_S to the unforced equilibrium point x_F. The minimum time optimal control problem is formulated as the following

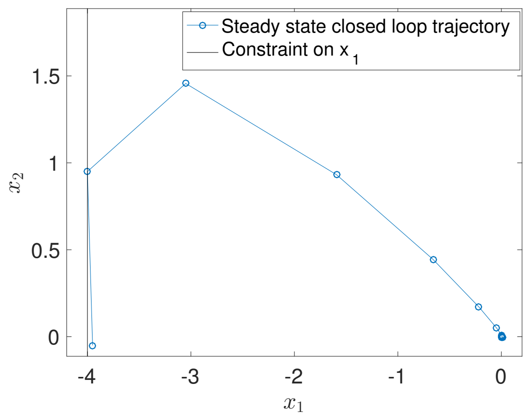

Fig. 2. The steady state trajectory saturates the state constraints.

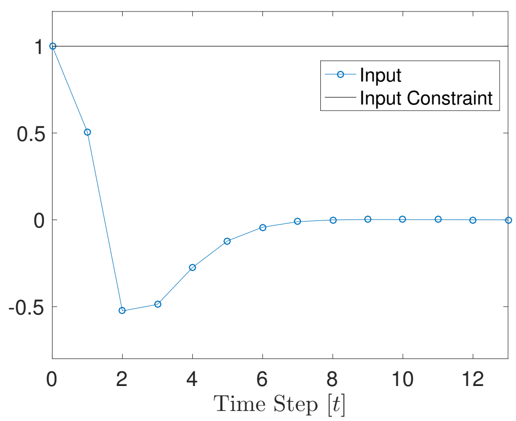

Fig. 3. The input associated with the steady state trajectory of the LMPC.

infinite time optimal control problem

$$J^*_{0\rightarrow\infty}(x_S) = \min_{\substack{\theta_0,\theta_1,\ldots \\ a_0,a_1,\ldots}} \sum_{k=0}^{\infty} \mathbb{1}_k \tag{64a}$$

s.t.

$$x_k = \begin{bmatrix} z_{k+1} \\ y_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} z_k \\ y_k \\ v_k \end{bmatrix} + \begin{bmatrix} v_k cos(\theta_k) \\ v_k sin(\theta_k) \\ a_k \end{bmatrix}, \ \forall k \geq 0 \tag{64b}$$

$$x_0 = x_S = [0 \quad 0 \quad 0]^T, \tag{64c}$$

$$-s \leq a_k \leq s, \quad \forall k \geq 0 \tag{64d}$$

$$\frac{(\hat{z}_k - z_{obs})^2}{a_e^2} + \frac{(\_k - y_{obs})^2}{b_e^2} \geq 1, \quad \forall k \geq 0. \tag{64e}$$

where the indicator function in (64a) is defined as

$$\mathbb{1}_k = \begin{cases} 1, & \text{if } x_k \neq x_F \\ 0, & \text{if } x_k = x_F \end{cases}. \tag{65}$$

In Equation (64d), $s = 1$ is the known acceleration saturation limit. Equations (64b)-(64c) represent the dynamic constraint and the initial conditions, respectively. The state vector $x_k = [z_k, y_k, v_k]$ collects the car's position on the $Z - Y$ plane and the velocity, respectively. The inputs are the heading angle, θ_k, and the acceleration command, a_k. Finally, (64e) represents the obstacle constraint, enforcing the system trajectory to lie outside the ellipse centered at (z_{obs},y_{obs}).

In order to find a local optimal solution to Problem (64), we implemented the LMPC (21) and (23) with the running cost $h(x_k, u_k) = \mathbb{1}_k$ and constraints (64b)-(64e). We set $x_F = [54, 0, 0]^T$, $a_e = 8$ and $b_e = 6$. At the 0-th iteration, we computed a feasible trajectory that steers system (64) from x_0 to x_F using a brute force algorithm. For efficient techniques to compute collision-free trajectories in the presence of obstacle we refer to [25], [26] and [27]. The feasible trajectory is used to construct the sampled safe set $\mathcal{SS}^0$, and the terminal cost, $Q^0(\cdot)$, needed to initialize the first iteration of the LMPC (21) and (23).

For this example, Problem (21) can be reformulated as a Mixed-Integer Quadratic Program (MIQP). Further details on its solution can be found in Section VII.A.2.

After 4 iterations the LMPC (21) and (23) converges to the steady state solution shown in Figure 4. Table IV shows that the cost is decreasing until convergence is achieved. The steady state inputs are reported in Figure 5, we notice that the controller saturates the acceleration as we would expect from the optimal solution to the minimum time Problem (64). In particular, the LMPC (21) and (23), similarly to a bang-bang [28] controller, accelerates until it reaches the midpoint between the initial and final position and afterwards it decelerates to reach the x_F with zeros velocity, as shown in Figure 6.

We performed an additional step to verify the (local) optimality of the steady state trajectory $\mathbf{x}^\infty$ at which the LMPC (21) and (23) converged. We solved problem (64) with an horizon $N = 16$ by using an interior point nonlinear solver [29] initialized with the trajectory obtained with our proposed approach at steady-state. We confirmed that the locally optimal solution of the solver coincides with the steady state solution of the LMPC (21) and (23).

TABLE IV

COST OF THE LMCPC AT EACH j-TH ITERATION

Iteration	Iteration Cost
$j = 0$	39
$j = 1$	21
$j = 2$	18
$j = 3$	17
$j = 4$	16
$j = 5$	16

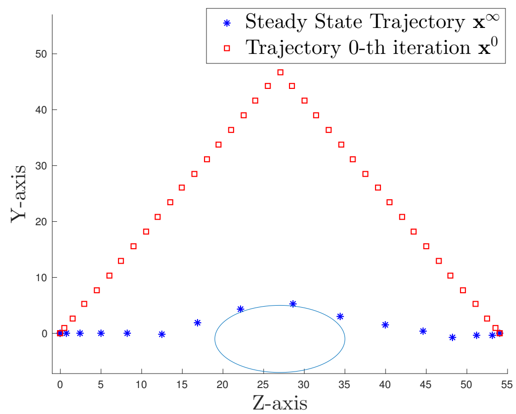

Fig. 4. Comparison between the first feasible trajectory $\mathbf{x}^0$ and the steady state trajectory $\mathbf{x}^\infty$.

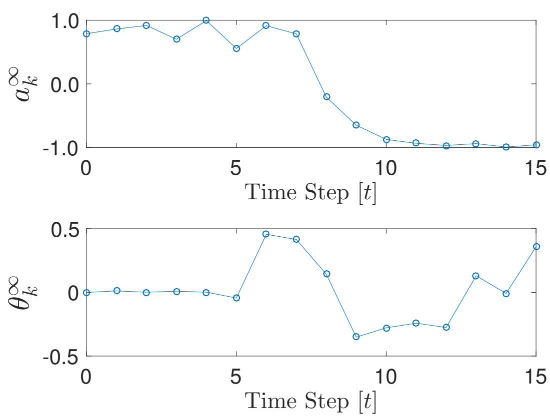

Fig. 5. The acceleration a_k^∞ and steering θ_k^∞ inputs associated with the steady state trajectory $\mathbf{x}^\infty$.

C. Dubins Car with Obstacle and Unknown Acceleration Saturation

Consider the minimum time Dubins car problem (64) presented in the previous example. We assume in this section that the saturation limit s is unknown. We use a sigmoid

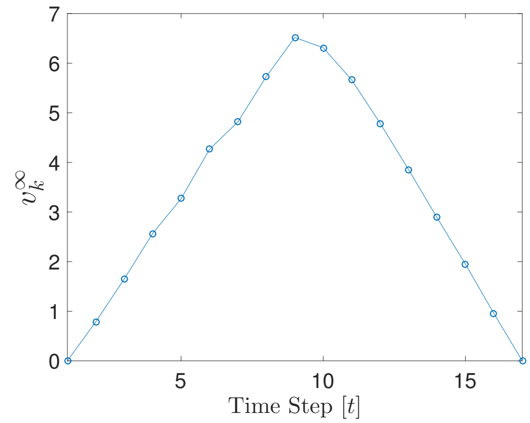

Fig. 6. The velocity profile v_k^∞ of the steady state trajectory $\mathbf{x}^\infty$.

function $\frac{a_k}{\sqrt{1+a_k^2}}$ as a continuously differentiable approximation of the saturation function and reformulate (64) as

$$J^*_{0\to\infty}(x_S) = \min_{\substack{\theta_0,\theta_1,\dots\\ a_0,a_1,\dots}} \sum_{k=0}^{\infty} \mathbb{1}_k \tag{66a}$$

s.t.

$$x_k = \begin{bmatrix} z_{k+1} \\ y_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} z_k \\ y_k \\ v_k \end{bmatrix} + \begin{bmatrix} v_k cos(\theta_k) \\ v_k sin(\theta_k) \\ s\frac{a_k}{\sqrt{1+a_k^2}} \end{bmatrix}, \ \forall k \geq 0 \tag{66b}$$

$$x_0 = x_S = [0 \ \ 0 \ \ 0]^T, \tag{66c}$$

$$\frac{(z_k - z_{obs})^2}{a_e^2} + \frac{(y_k - y_{obs})^2}{b_e^2} \geq 1, \quad \forall k \geq 0. \tag{66d}$$

where the indicator functino $\mathbb{1}_k$ is defined in (65). The state vector $x_k = [z_k, y_k, v_k]$ collects the car position of car on the Z-Y plane and the velocity, respectively. The inputs are the acceleration a_k and the heading angle θ_k. Finally, s represents the unknown saturation limit. As in the previous example, we set $x_F = [54, 0, 0]^T$, $a_e = 8$ and $b_e = 6$. The vehicle model uses a saturation limit $s = 1$. This is unknown to the controller.

We apply the proposed LMPC on an augmented system to simultaneously estimated the saturation coefficient and to steer the system (66b) to the terminal point x_F. In order to archive this, we define a saturation coefficient estimate, $\hat{s}_k$, and an error estimate $e_k = s - \hat{s}_k$. The idea of augmenting the system with an estimator and a related error dynamics is standard in adaptive control strategies [30] [31]. The objective of the controller is a trade off between estimating the saturation coefficient and steering the system to the terminal point x_F. The LMPC solves at time t of the j-th iteration the following problem,

$$J^{\text{LMPC},j}_{0\rightarrow N}(x^j_t) = \min_{\substack{\theta_0,\ldots,\theta_N \\ a_0,\ldots,a_N \\ \delta_0,\ldots,\theta_N}} \left[\sum_{k=0}^{N-1} w_e e_k^2 + \bar{\mathbb{1}}_k \right] + Q^{j-1}(x_N) \tag{67a}$$

s.t.

$$\hat{x}_{k+1} = \hat{f}(\hat{x}_k, \hat{u}_k) = \begin{bmatrix} \hat{z}_k \\ \hat{y}_k \\ \hat{v}_k \\ \hat{s}_k \\ e_k \end{bmatrix} + \begin{bmatrix} \hat{v}_k cos(\theta_k) \\ \hat{v}_k sin(\theta_k) \\ \hat{s}_{k+1} \frac{a_k}{\sqrt{1+a_k^2}} \\ \delta_k \\ -\delta_k \end{bmatrix}, \ \forall k \geq 0 \tag{67b}$$

$$\hat{x}_0 = x^j_t, \tag{67c}$$

$$\frac{(\hat{z}_k - z_{obs})^2}{a_e^2} + \frac{(\hat{y}_k - y_{obs})^2}{b_e^2} \geq 1, \quad \forall k \geq 0, \tag{67d}$$

$$x_N \in \ \mathcal{SS}^{j-1}, \tag{67e}$$

where $N = 4$ and the weight on the error estimate $w_e = 10$. The indicator function $\bar{\mathbb{1}}_k$ in (67a) is defined as

$$\bar{\mathbb{1}}_k = \begin{cases} 1, & \text{if } \hat{x}_k \notin \mathcal{X}_F \\ 0, & \text{if } \hat{x}_k \in \mathcal{X}_F \end{cases}. \tag{68}$$

where

$$\mathcal{X}_F = \left\{ \bar{x} = \begin{bmatrix} \hat{z} \\ \hat{y} \\ \hat{v} \\ \hat{s} \\ e \end{bmatrix} \in \mathbb{R}^5 : \begin{bmatrix} \hat{z} \\ \hat{y} \\ \hat{v} \end{bmatrix} = x_F, \hat{s} \in \mathbb{R}, e = 0 \right\}. \tag{69}$$

$\hat{f}(\cdot,\cdot)$ in (67b) represents the dynamics update of the augmented system and the state vector $\hat{x}_k = [\hat{z}_k, \hat{y}_k, \hat{v}_k, \hat{s}_k, e_k]$ collects the estimate position on the Z-Y plane, the car's velocity, the saturation coefficient estimator and the estimator error, respectively. The input vector $\hat{u}_k = [a_k, \theta_k, \delta_k]$ collects the acceleration, the steering and the estimate difference between two consecutive time steps, respectively. Equation (67c) represents the initial condition and (67d) the obstacle avoidance constraint. Constraint (67e) enforces the terminal state into the $\mathcal{SS}^{j-1}$ defined in equation (12). Finally, in (67) we have used a simplified notation to equation (21).

Let at time t of the j-th iteration $\mathbf{u}^{*,j}_{t:t+N|t}$ be the optimal solution to (67), then we apply the first element of $\mathbf{u}^{*,j}_{t:t+N|t}$ to the system in (67b)

$$u^j_t = u^{*,j}_{t|t}. \tag{70}$$

We assume that at time t of the j-th iteration the system state $x^j_t = [z^j_t, y^j_t, v^j_t]$ is measured and we estimate e^j_t inverting the system dynamics (66b) and (67b)

$$e^j_t = \begin{cases} \frac{y^j_t - \hat{y}^j_t - (y^j_{t-1} - \hat{y}^j_{t-1})}{\frac{a^j_{t-1}}{\sqrt{1+(a^j_{t-1})^2}}}, & \text{If } \frac{a_{t-1}}{\sqrt{1+a^2_{t-1}}} \neq 0 \\ e^j_{t-1} & \text{otherwise} \end{cases}. \tag{71}$$

Remark 9: Consider a local optimal solution $\bar{\mathbf{x}}^* = [\mathbf{z}^*, \ \hat{\mathbf{y}}^*, \ \mathbf{v}^*, \ \hat{\mathbf{s}}^*, \ \mathbf{e}^*]^T$ to problem (67) defined over the infinite horizon. If $e^*_k = 0, \ \forall k > 0$, (i.e., the algorithm has successfully identified the friction saturation coefficient), then $\mathbf{x}^* = [\mathbf{z}^*, \ \hat{\mathbf{y}}^*, \ \mathbf{v}^*]$ is a local optimal solution for the original problem (64).

Initialization of the LMPC (67) is discussed in the Appendix. For this example, Problem (21) can be reformulated as a Mixed-Integer Quadratic Program (MIQP). Further details on its solution can be found in Section VII.A.2.

After 7 iterations, the LMPC (67), (70) converges to a steady state solution. Figure 8 illustrates the evolution of the sampled safe set through the iterations and Table V shows that the iteration cost is decreasing until convergence is reached.

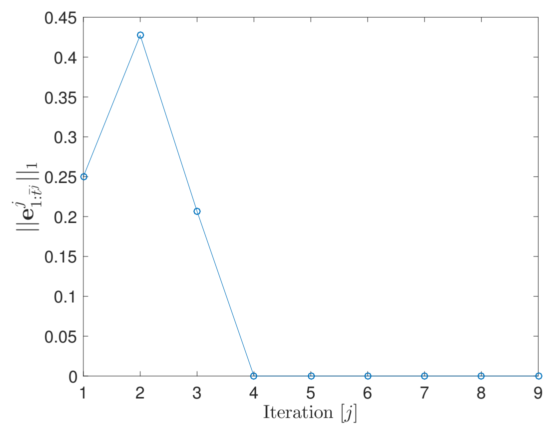

Fig. 7. Evolution of the 1-norm of the estimation error through the iterations.

Figure 7 shows the behavior of the 1-norm of the error vector

$$\mathbf{e}^j_{1:\infty} = [e^j_1, \ldots, e^j_t, \ldots]. \tag{72}$$

as a function of the iteration j. We notice that the LMPC (67), (70) correctly learns from the previous iterations decreasing the estimation error, until it identifies the unknown saturation coefficient (i.e. $e^\infty_k = 0 \ \forall k > 0$).

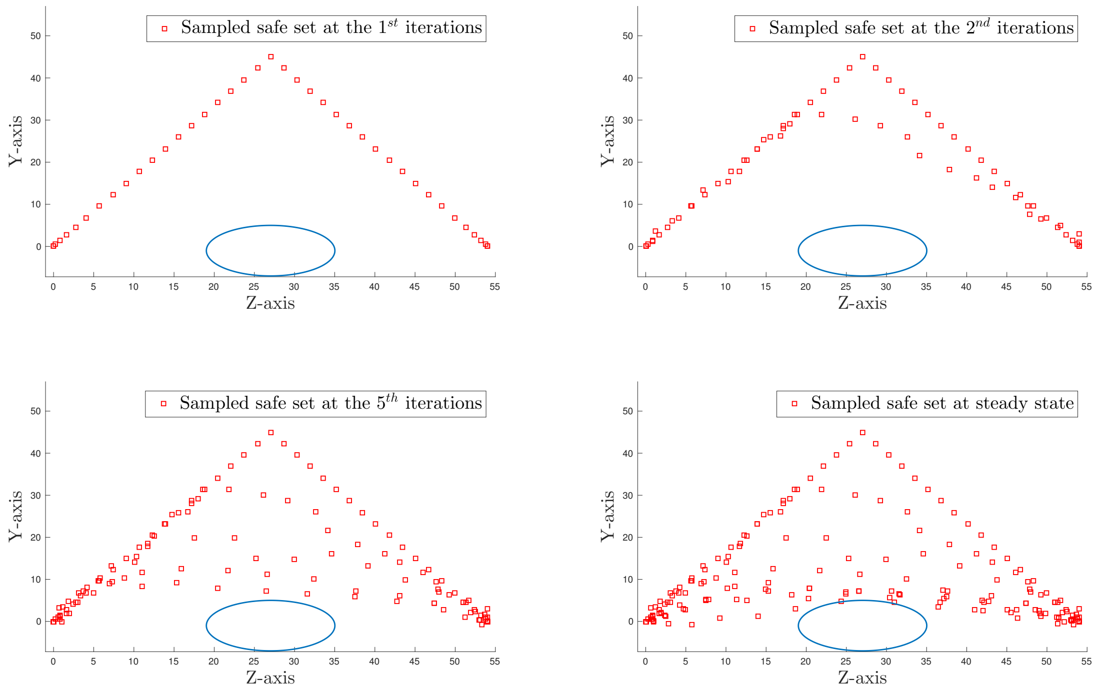

Fig. 8. Sampled safe set evolution over the iterations.

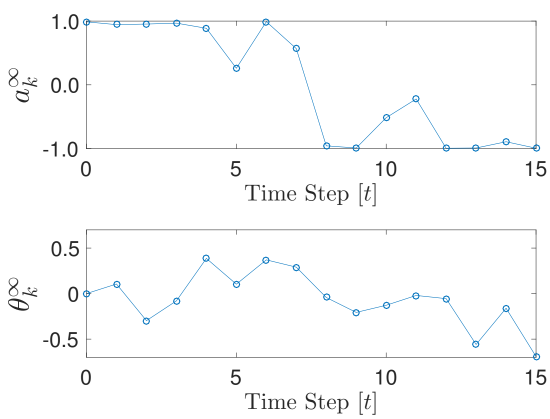

Fig. 9. The acceleration a_k^∞ and steering θ_k^∞ inputs associated with the steady state trajectory $\mathbf{x}^\infty$.

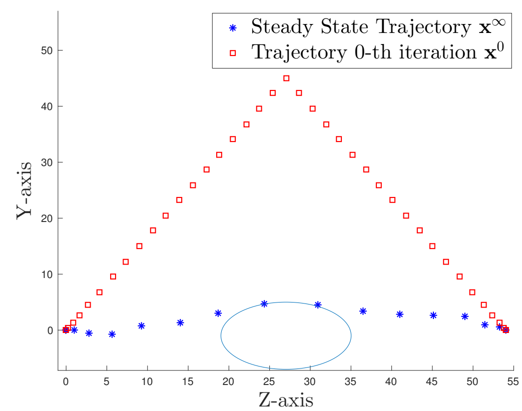

Fig. 10. Steady state trajectory of the LMPC on the $Z - Y$ plane.

The steady state inputs are reported in Figure 9. One can observe that the LMPC (67), (70) saturates the acceleration constraints. The controller accelerates until it reaches the midpoint between the initial and final position and it decelerates afterwards, as we would expect from the optimal solution to a minimum time problem [28]. Figure 10 shows the steady state trajectory $\mathbf{x}^\infty$, and the feasible trajectory

$\mathbf{x^0}$ at the 0-th iteration. The LMPC (67) and (70) steers the system from the staring point x_S to the final point x_F in 16 steps as the optimal solution to (64) computed in the previous example.

TABLE V

OPTIMAL COST OF THE LMPC AT EACH j-TH ITERATION

Iteration	Iteration Cost
$j=0$	65.000000000000000
$j=1$	33.634529488066327
$j=2$	24.216166714512450
$j=3$	19.625000000001727
$j=4$	19.625000000000004
$j=5$	17.625000000022546
$j=6$	17.625000000000000
$j=7$	16.625000000000000
$j=8$	16.625000000000000

V. PRACTICAL CONSIDERATIONS

A. Computation

The sampled safe set (12) is a set of discrete points and therefore the terminal constraint in (21e) is an integer constraint. Consequently, the proposed approach is computationally expensive also for linear system as the controller has to solve a mixed integer programming problem at each time step. In the following we discuss two different approaches to improve the computational burden associated with the proposed control logic.

1) Convexifing the terminal constraint: The computational burden associated with the finite time optimal control problem (21) can be reduced relaxing the sampled safe to its convex hull, and the $Q(\cdot)$ function to be its barycentric approximation. For more details on barycentric approximation we refer to [17]. This relaxed problem is convex if the system dynamics is linear and the stage cost is convex. Furthermore, for linear system and convex stage cost, the relaxed approach preserves the properties showed in *Theorems 1-3* of [32]. When the system is nonlinear, it is still possible to apply the convex relaxation but guarantees are, in general, lost. In [33], this relaxed approach has been successfully applied in real time to the nonlinear minimum time autonomous racing problem, where the LMPC is used to improve the vehicle's lap time over the iterations. A video of a more recent implementation on the Berkeley Autonomous Racing Car (BARC) platform can be found here: `https://automatedcars.space/home/2016/12/22/learning-mpc-for-autonomous-racing`.

2) Parallelize Computations: The structure of the LMPC can be exploited to design an algorithm that: *i*) use a subset of the sampled safe in the (21), *ii*) can be parallelized. In particular, one can compute an upper and lower bound to the optimal solution of problem (21). These bounds allow to reduce the complexity of (21) without loosing the guarantees proven in *Theorems 1-3*. More details are discussed next.

First, we notice that at time t, $\forall t > 0$ it is possible to compute an upper bound on the optimal cost of problem (21), using the solution computed at time $t-1$. In particular, from equations (6) and (33) we have,

$$J^{\text{LMPC},j}_{0\to N}(x^j_t) - J^{\text{LMPC},j}_{0\to N}(x^j_{t-1}) \leq -h(x^j_{t-1}, u^j_{t-1}) \leq 0, \quad (73)$$

which implies that at time t an upper bound on the optimal cost is given by

$$J^{\text{LMPC},j}_{0\to N}(x^j_t) \leq J^{\text{LMPC},j}_{0\to N}(x^j_{t-1}). \quad (74)$$

In order to compute a lower bound, let (22) be the optimal solution to (21), then at the j-th iteration

$$J^{\text{LMPC},j}_{t\to t+N}(x^j_t) = \sum_{k=t}^{t+N-1} h(x^{*,j}_{k|t}, u^{*,j}_{k|t}) + Q^{j-1}(x^{*,j}_{t+N|t}). \quad (75)$$

As Problem (67) is time-invariant and $h(\cdot,\cdot)$ is positive definite (6), we have

$$J^{\text{LMPC},j}_{0\to N}(x^j_t) \geq Q^{j-1}(x^{*,j}_{t+N|t}), \quad \forall x^{*,j}_{t+N|t} \in \mathcal{SS}^{j-1}. \quad (76)$$

Combining the upper bound (74) and the lower bound (76), we obtain

$$Q^{j-1}(x^{*,j}_{t+N|t}) \leq J^{\text{LMPC},j}_{0\to N}(x^j_t) \leq J^{\text{LMPC},j}_{0\to N}(x^j_{t-1}). \quad (77)$$

Therefore at optimum we have that

$$Q^{j-1}(x^{*,j}_{t+N|t}) \leq J^{\text{LMPC},j}_{0\to N}(x^j_{t-1}). \quad (78)$$

Define $\mathcal{RS}^{j-1}_t$ as the set of points which satisfy condition (78),

$$\mathcal{RS}^{j-1}_t = \{x \in \mathcal{SS}^{j-1}_t : Q^{j-1}(x) \leq J^{\text{LMPC},j}_{0\to N}(x^j_{t-1})\}, \quad (79)$$

then, from equation (78), we deduce that for $t > 0$

$$x^{*,j}_{t+N|t} \in \mathcal{RS}^{j-1}_t \subseteq \mathcal{SS}^{j-1}. \quad (80)$$

The set $\mathcal{RS}$ can be used in place of $\mathcal{SS}$ in order to reduce computational complexity.

The following **Algorithm 1** uses this idea to solve the LMPC (21), (23). **Algorithm 1** was used for the Dubins Car example with the nonlinear solver Ipopt [29].

Algorithm 1: Compute u_t^j at time t of the j-th iteration

```
Read measurements and update x_t^j and t.
if t > 0 then
    Compute RS_t^{j-1}
else
    Set RS_t^{j-1} = SS^{j-1}
end
n = 0
for all x ∈ RS_t^{j-1} do
    In (21), set SS^{j-1} = x
    Solve (21) using a nonlinear optimization solver.
    Set ū_n = u*,j_{t|t} and J̄_n = J^{LMPC,j}_{t→t+N}(x_t^j)
    n = n + 1
end
Find n* = argmin_n J̄_n
Apply u_t^j = ū_{n*}
```

B. Uncertainty

The paper uses a deterministic framework and the theoretical guaranties have been demonstrated only for the deterministic case. This is the case of the vast majority of seminal papers on MPC [16], [34], [35], [36]. In the presence of disturbances, as for all deterministic MPC schemes, all the guarantees are lost. However, one can build on the proposed results to formulate a *stochastic iterative learning MPC*. For instance if disturbance is modeled as a Gaussian process the chance constraint can be converted to deterministic second order cone constraint [37], which can be handled with the proposed control logic. Furthermore, the proposed control logic can be extended to a *robust iterative learning MPC* when the disturbance is bounded and the system is linear. Under these assumptions the robust MPC can be formulated in a deterministic control problem tightening the constraints [38]. In particular, the robust MPC can be designed on a nominal model where the tightening of the state constraints is computed to guarantee that the original system satisfies the nominal constraints for all the disturbance values [38]. This is topic of further investigation.

VI. Conclusions

In this paper, a reference-free learning nonlinear model predictive control that exploits information from the previous iterations to improve the performance of the closed loop system over iterations is presented. A safe set and a terminal cost, learnt from previous iterations, allow to guarantee the recursive feasibility and stability of the closed loop system. Moreover, we showed that if the closed-loop system converges to steady state trajectory then this trajectory is locally optimal for the infinite horizon optimal control problem, regardless of the LMPC optimization horizon. We tested the proposed control logic on an infinite horizon linear quadratic regulator with constraints (CLQR) to shown that the proposed control logic converges to the optimal solution of the infinite optimal control problem. Finally, we tested the control logic on nonlinear minimum time problem optimal control problem and we showed that the properties of the proposed LMPC can be used to simultaneously estimate unknown system parameters and to generate a state trajectory that pushes system performance.

VII. Acknowledgments

We thank the reviews for their feedback on the manuscript.

VIII. Appendix

In order to compute a feasible trajectory that steers system (67b) from the initial state $\bar{x}_0 = [x_0, \hat{s}_0, e_0]^T$ into $\mathcal{X}_F$ we used a greedy approach described next. First, we set $\delta_k = 0,\ \forall k = 1, \cdots, N-1$. Therefore, from (67b), we have that

$$\hat{s}_k = \hat{s}_0,\ \forall k = 1, \cdots, N-1 \tag{81a}$$
$$e_k = e_0,\ \forall k = 1, \cdots, N-1. \tag{81b}$$

Afterwards, we selected an initial guess for the saturation coefficient estimate $\hat{s}_0 = 0.25$ and given the following input structure

$$\theta_k = \tilde{\theta}, \qquad \forall k = 1, \cdots, N_s \tag{82a}$$
$$\theta_k = -\tilde{\theta}, \qquad \forall k = N_s + 1, \cdots, N-1 \tag{82b}$$
$$a_k = \tilde{a}, \qquad \forall k = 1, \cdots, \bar{N}_s \tag{82c}$$
$$a_k = 0, \qquad \forall k = \bar{N}_s + 1, \cdots, N - \bar{N}_s \tag{82d}$$
$$a_{N-1} = -\tilde{a}, \qquad \forall k = N - \bar{N}_s + 1, \cdots, N \tag{82e}$$

we generated a set of trajectories using different sets of parameters $\tilde{\theta}, N_s, \bar{N}_s, \tilde{a}, N$. Among the generated trajectories, we used the one minimizing the following quantity

$$|| \begin{bmatrix} z_{N-1} \\ \bar{y}_{N-1} \\ v_{N-1} \\ \hat{s}_{N-1} \\ e_{N-1} \end{bmatrix} - \begin{bmatrix} x_F \\ \hat{s}_{N-1} \\ e_{N-1} \end{bmatrix} ||_2^2 \tag{83}$$

to warm-start a nonlinear optimization problem which allowed us to find the following $N-1$ step trajectory

$$\bar{\mathbf{x}}^0_{0:N-1} = \Big[\bar{x}_0^0, \cdots, \bar{x}^0_{N-1} = \begin{bmatrix} x_F \\ \hat{s}_{N-1} \\ e_{N-1} \end{bmatrix}\Big], \tag{84}$$

and the related input sequence

$$(\theta_k^0, a_k^0),\ \forall k = 1, \cdots, N-1. \tag{85}$$

Afterwards the input sequence (85) are applied to the system (64b) to compute

$$\mathbf{x}^0_{0:N-1} = [x^0_0, \cdots, x^0_{N-1}]. \tag{86}$$

Then realized trajectories $\bar{\mathbf{x}}^0_{0:N-1}$ and $\mathbf{x}^0_{0:N-1}$ are used to compute the error, which from equations (64b) and (67b), is given by

$$e_{k+1} = \begin{cases} \frac{y_{k+1} - \hat{y}_{k+1} - (y_k - \hat{y}_k)}{\frac{a_k}{\sqrt{1+a_k^2}}}, & \text{If } \frac{a_k}{\sqrt{1+a_k^2}} \neq 0 \\ e_k & \text{else} \end{cases} \tag{87}$$

$\forall k = 0, \cdots, N-2.$

Finally, we selected

$$\theta^0_N = a^0_N = 0 \tag{88a}$$
$$\delta^0_N = e_{N-1} \tag{88b}$$

to regulate e^0_{N-1} to zero steering $\bar{x}^0_{N-1}$ into $\mathcal{X}_F$. Concluding, the N steps trajectory which extends the trajectory in (84) using (88),

$$\bar{\mathbf{x}}^0_{0:N} = \left[\bar{x}^0_0, \cdots, \bar{x}^0_N = \begin{bmatrix} x_F \\ \hat{s}_N \\ 0 \end{bmatrix}\right] \tag{89}$$

steers system (67b) into $\mathcal{X}_F$ and it can be used to build $\mathcal{SS}^0$ and $Q^0(\cdot)$.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.

[2] 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.

[3] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Engineering Practice*, vol. 15, no. 10, pp. 1306–1318, 2007.

[4] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[5] C.-Y. Lin, L. Sun, and M. Tomizuka, "Matrix factorization for design of q-filter in iterative learning control," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 6076–6082.

[6] ——, "Robust principal component analysis for iterative learning control of precision motion systems with non-repetitive disturbances," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2819–2824.

[7] K. S. Lee, I.-S. Chin, H. J. Lee, and J. H. Lee, "Model predictive control technique combined with iterative learning for batch processes," *AIChE Journal*, vol. 45, no. 10, pp. 2175–2187, 1999.

[8] 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.

[9] 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.

[10] J. R. Cueli and C. Bordons, "Iterative nonlinear model predictive control. stability, robustness and applications," *Control Engineering Practice*, vol. 16, no. 9, pp. 1023–1034, 2008.

[11] R. Sharp and H. Peng, "Vehicle dynamics applications of optimal control theory," *Vehicle System Dynamics*, vol. 49, no. 7, pp. 1073–1111, 2011.

[12] A. Rucco, G. Notarstefano, and J. Hauser, "An efficient minimum-time trajectory generation strategy for two-track car vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1505–1519, 2015.

[13] Y.-L. Hwang, T.-N. Ta, C.-H. Chen, and K.-N. Chen, "Using zero moment point preview control formulation to generate nonlinear trajectories of walking patterns on humanoid robots," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*. IEEE, 2015, pp. 2405–2411.

[14] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2589–2594.

[15] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[17] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Springer, 2003, vol. 290.

[18] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[19] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[20] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[21] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya *et al.*, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008.

[22] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[23] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME 2010 dynamic systems and control conference*. American Society of Mechanical Engineers, 2010, pp. 265–272.

[24] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011.

[26] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using rrt," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 1681–1686.

[27] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, 2010.

[28] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.

[29] H. Pirnay, R. López-Negrete, and L. T. Biegler, "Optimal sensitivity based on ipopt," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 307–331, 2012.

[30] H. Bai, M. Arcak, and J. T. Wen, "Adaptive motion coordination: Using relative velocity feedback to track a reference velocity," *Automatica*, vol. 45, no. 4, pp. 1020–1025, 2009.

[31] G. C. Goodwin, R. L. Leal, D. Q. Mayne, and R. H. Middleton, "Rapprochement between continuous and discrete model reference adaptive control," *Automatica*, vol. 22, no. 2, pp. 199–207, 1986.

[32] U. Rosolia and F. Borrelli, "Learning Model Predictive Control for Iterative Tasks: A Computationally Efficient Approach for Linear System," *ArXiv e-prints*, Feb. 2017.

[33] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous Racing using Learning Model Predictive Control," *ArXiv e-prints*, Oct. 2016.

[34] A. Zheng and M. Morari, "Stability of model predictive control with mixed constraints," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1818–1823, 1995.

[35] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[36] S. L. de Oliveira Kothare and M. Morari, "Contractive model predictive control for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1053–1071, 2000.

[37] G. C. Calafiore and L. El Ghaoui, "Linear programming with probability constraints-part 1," in *American Control Conference, 2007. ACC'07.* IEEE, 2007, pp. 2636–2641.

[38] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Springer, 2015.

Ugo Rosolia received the B.S. and M.S. (*cum laude*) degrees in mechanical engineering from the Politecnico di Milano, Milan, Italy, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the University of California at Berkeley, Berkeley, CA, USA.

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.

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.