2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Repetitive Learning Model Predictive Control: An Autonomous Racing Example

Maximilian Brunner, Ugo Rosolia, Jon Gonzales, and Francesco Borrelli

Abstract— **We propose an optimization based, data-driven framework to design controllers for repetitive tasks. The proposed framework builds on previous work of Learning Model Predictive Control and focuses on problems where the terminal condition of one iteration is the initial condition of the next iteration. A terminal cost and a sampled safe set are learned from data to guarantee recursive feasibility and non-decreasing performance cost at each iteration. The proposed control logic is tested on an autonomous racing example, where the vehicle dynamics are identified online. Experimental results on a 1:10 scale RC car illustrate the effectiveness of the proposed approach.**

I. Introduction

Over the last decades research has been focusing on autonomous driving. Several techniques have been proposed for highway scenarios [1], [2]. In these situations the vehicle does not operate close to the limit of its handling capability and its dynamics can be approximated using simplified models [3], [4]. In this work we are interested in designing a controller for autonomous racing which can operate the vehicle close to the limit of its handling capability while guaranteeing safety. We propose to formulate the autonomous racing problem as a repetitive control task. At each iteration the controller drives the vehicle around the track trying to minimize the lap time. The data of each iteration is stored and used to improve the system performance in future iterations.

Different approaches have been proposed to tackle the problem of autonomous racing. In [5] a cascaded MPC control system is proposed. A high level MPC computes an optimal racing trajectory, a low level MPC ensures tracking of this trajectory. The authors in [5] also proposed a single layer Model Predictive Contouring Control (MPCC), where the controller objective is a trade-off between the progress along the track and the contouring error. In [6], the authors compared two approaches, the first one based on a tracking MPC, and, the second one, based on an MPC formulated in a space dependent frame. An iterative learning control (ILC) approach for autonomous racing has been proposed in [7]. The ILC framework is used to reduce the tracking error from a reference trajectory. This reference trajectory is computed offline using the method presented in [8]. The authors showed the effectiveness of the proposed ILC with experimental testing on a full size vehicle. In [9] we proposed to solve the autonomous racing problem as an iterative control problem.

The main contribution of this paper is to extend the formulation proposed in [9] to deal with repetitive tasks. The proposed Repetitive Learning MPC does not need to start from the same initial condition at each iteration and it can be used to improve the performance of a vehicle running continuously on a race track. In particular, we show that our new formulation guarantees that *(i):* the j-th iteration cost is non-increasing over the iterations, *(ii):* state and input constraints are satisfied at iteration j if they were satisfied at iteration $j-1$, *(iii):* if the controller goal is to steer the system to a control invariant set $\mathcal{X}_\mathcal{F}$, then the closed loop system asymptotically converges to this set.

The second contribution of this work is to test the controller on an experimental set-up. In order to achieve this goal a complex nonlinear model able to capture the vehicle lateral dynamics is needed. However, even for non aggressive maneuvers in which the linear tire model assumption holds, the tire stiffness is speed and environment dependent [10]. Therefore, we decided to run a least mean square algorithm in real time to identify the vehicle parameters which change over time and space. The idea of using a regressor to identify the system dynamics used in the MPC controller is a well-established technique in literature and it has been successfully used in different applications [11], [12].

This paper is organized as follows: in Section II we introduce the minimum time problem. Section III gives a brief overview over the notation used in LMPC. The LMPC for repetitive tasks is presented in Section IV. In Section V we show experimental results on a 1:10 scale race car. Section VI provides final remarks.

II. Problem Definition

A. Vehicle Model

The system model is concisely expressed in state space representation as

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

where at time t the vectors x_t and u_t collect the states and input,

$$x_t = [v_{x_t}\ v_{y_t}\ \dot{\psi}_t\ e_{\psi_t}\ e_{y_t}\ s_t] \text{ and } u_t \ = [a_t\ \delta_t]. \tag{2}$$

The quantities v_{x_t}, v_{y_t}, and $\dot{\psi}_t$ represent the vehicle longitudinal velocity, lateral velocity, and yaw rate, respectively. s_t is the distance travelled along the centerline of the road, e_{ψ_t} and e_{y_t} are the heading angle and lateral distance

Maximilian Brunner is with the Department of Mechanical Engineering, ETH Zuerich, Zuerich 8092, Switzerland {mabrunne} @ student.ethz.ch

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

error between the vehicle and the path. The inputs are the longitudinal acceleration a_t and the steering angle δ_t. The system dynamics $f(x_t, u_t)$ in (1) are given by an Euler discretized dynamic bicycle model [3], [4],

$$f(x_t,u_t) = \\ = \begin{bmatrix} \theta^{(1)}_{v_x} v_{x_t} + \theta^{(2)}_{v_x} \dot{\psi}_t v_{y_t} + \theta^{(3)}_{v_x} a_t \\ v_{y_t} + \theta^{(1)}_{v_y} \frac{v_{y_t}}{v_{x_t}} + \theta^{(2)}_{v_y} \frac{\dot{\psi}_t}{v_{x_t}} + \theta^{(3)}_{v_y} \dot{\psi}_{y_t} v_{x_t} + \theta^{(4)}_{v_y} \delta_t \\ \dot{\psi}_t + \theta^{(1)}_{\dot{\psi}} \frac{v_{y_t}}{v_{x_t}} + \theta^{(2)}_{\dot{\psi}} \frac{\dot{\psi}_t}{v_{x_t}} + \theta^{(3)}_{\dot{\psi}} \delta_t \\ e_{\psi_t} + \Delta t \Big(\dot{\psi}_t - \frac{v_{x_t} \cos(e_{\psi_t}) - v_{y_t} \sin(e_{\psi_t})}{1 - e_{y_t} c(s_t)} c(s_t) \Big) \\ e_{y_t} + \Delta t (v_{x_t} \sin(e_{\psi_t}) + v_{y_t} \cos(e_{\psi_t})) \\ s_t + \Delta t \frac{v_{x_t} \cos(e_{\psi_t}) - v_{y_t} \sin(e_{\psi_t})}{1 - e_{y_t} c(s_t)} \end{bmatrix} \tag{3}$$

where $c(\cdot)$ is the road curvature, Δt the discretization step and I_Z and m the moment of inertia and mass of the car, respectively. The coefficients θ_{v_y} and $\theta_{\dot{\psi}}$ describing the lateral dynamics in (3) are given by

$$\begin{aligned} &\theta^{(1)}_{v_y} = \frac{C_{\alpha f} + C_{\alpha r}}{m} \Delta t, &&\theta^{(2)}_{v_y} = \frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{m} \Delta t \\ &\theta^{(3)}_{v_y} = -\Delta t, &&\theta^{(4)}_{v_y} = \frac{C_{\alpha f}}{m} \Delta t \\ &\theta^{(1)}_{\dot{\psi}} = \frac{l_f C_{\alpha f} - l_r C_{\alpha r}}{I_z} \Delta t, &&\theta^{(2)}_{\dot{\psi}} = \frac{l_f^2 C_{\alpha f} + l_r^2 C_{\alpha r}}{I_z} \Delta t \\ &\theta^{(3)}_{\dot{\psi}} = \frac{l_f C_{\alpha f}}{I_z} \Delta t \end{aligned} \tag{4}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ represent the front and rear cornering stiffness. l_f and l_r are the distances from the center of gravity to the front and rear axles, respectively.

The vector of coefficients θ_{v_x} for the longitudinal dynamic in (3) is given by

$$\theta^{(1)}_{v_x} = 1, \quad \theta^{(2)}_{v_x} = \Delta t, \quad \theta^{(3)}_{v_x} = \Delta t. \tag{5}$$

In general, the coefficients describing the lateral dynamics in (4) are time and space dependent [10]. Moreover, the coefficients in (5) depend on the motor dynamics which change at different speeds. Therefore, we used a simple online Least Mean Squares (LMS) algorithm to recursively estimate the coefficients in (4)-(5). We refer to [9] for further implementation details.

B. Control Problem

The goal of the controller is to minimize the lap time. The minimum time optimal control problem can be restated as the following equivalent infinite horizon optimal control problem

$$J^*_{0\to\infty} = \min_{x_0, u_0, u_1, \dots} \sum_{k=0}^{\infty} h(x_k, u_k) \tag{6a}$$
$$\text{s.t. } x_{k+1} = f(x_k, u_k), \ \forall k \geq 0 \tag{6b}$$
$$x_0 = \mathcal{L} \tag{6c}$$
$$x_k \in \mathcal{X}, \ u_k \in \mathcal{U}, \ \forall k \geq 0 \tag{6d}$$

where (6d) are the state and input constraints. (6b) constrains the state to evolute according to the system model. The constraint (6c) enforces the initial position of the vehicle to be on the starting line of the track

$$\mathcal{L} = \Big\{ x \in \mathbb{R}^6 : \ x^T e_6 = 0 \Big\} \tag{7}$$

where e_6 is the the 6^{th} standard basis in $\mathbb{R}^6$.
The stage cost $h(\cdot, \cdot)$ in (6a) is defined as

$$h(x_t, u_t) = \begin{cases} 1 & \text{if } x_t \notin \mathcal{X}_{\mathcal{F}} \\ 0 & \text{if } x_t \in \mathcal{X}_F \end{cases}, \tag{8}$$

where $\mathcal{X}_F$ is the set of points beyond the finish line at the end of the track of length s_F,

$$\mathcal{X}_F = \Big\{ x \in \mathbb{R}^6 : \ x^T e_6 = s > s_F \Big\}. \tag{9}$$

Assumption 1: We assume that after the vehicle has crossed the finish line there exists a controller that can keep the vehicle on the road. Namely, we assume that $\mathcal{X}_F$ is a control invariant set, thus $\forall x_k \in \mathcal{X}_F, \exists u_k \in \mathcal{U} : x_{k+1} = f(x_k, u_k) \in \mathcal{X}_F$.

Remark 1: We underline that (6) is not a classical optimal control problem as the initial condition is an optimization variable. We defined x_0 as an optimization variable to allow the controller to pick the best initial position on the finish line to perform a fast lap.

III. LMPC Preliminaries

We propose to reformulate the autonomous racing problem as repetitive task, where at each j-th iteration the controller uses the data from the previous j-1 iterations to improve the vehicle's lap time. At each j-th iteration the initial condition is given by the final condition of the previous iteration, thus this new formulation allows the controller to run continuously on a race track.

In the following, we introduce the notation that will be used to implement the Repetitive Learning MPC for the racing application.

A. Sampled Safe Set

At the j-th iteration the vectors

$$\mathbf{u}^j = [u_0^j, \ u_1^j, \ ..., \ u_t^j, \ ...], \tag{10a}$$
$$\mathbf{x}^j = [x_0^j, \ x_1^j, \ ..., \ x_t^j, \ ...], \tag{10b}$$

collect the inputs applied to vehicle model (1) and the corresponding state, where x_t^j and u_t^j denote the vehicle state and the control input at time t of the j-th iteration.

We define the the *sampled Safe Set* $\mathcal{SS}^j$ at iteration j as

$$\mathcal{SS}^j = \Bigg\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{\infty} x_t^i \Bigg\}, M^j = \Big\{ k \in [0, j] : \lim_{t\to\infty} x_t^k \in \mathcal{X}_F \Big\}. \tag{11}$$

$\mathcal{SS}^j$ is the collection of all state trajectories at iteration i for $i \in M^j$. M^j in equation (11) is the set of indexes k associated with successful iterations k for $k \leq j$.

B. Iteration Cost

At time t of the j-th iteration the cost-to-go associated with the closed loop trajectory (10b) and input sequence (10a) is defined as

$$J^j_{t\to\infty}(x^j_t) = \sum_{k=t}^{\infty} h(x^j_k, u^j_k), \tag{12}$$

where $h(\cdot,\cdot)$ is the stage cost of the problem (6). We define the j-th iteration cost as the cost (12) of the j-th trajectory at time $t=0$. $J^j_{0\to\infty}(x^j_0)$ quantifies the controller performance at each j-th iteration.

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

$$Q^j(x) = \begin{cases} \min\limits_{(i,t)\in F^j(x)} J^i_{t\to\infty}(x), & \text{if } x \in \mathcal{SS}^j \\ +\infty, & \text{if } x \notin \mathcal{SS}^j \end{cases}, \tag{13}$$

where $F^j(\cdot)$ in (13) is defined as

$$F^j(x) = \Big\{(i,t) : i \in [0,j],\ t \geq 0 \text{ with } x = x^i_t; \text{ for } x^i_t \in \mathcal{SS}^j\Big\}. \tag{14}$$

For more details on the LMPC notation we refer to [13].

IV. Repetitive LMPC Control Design

A. LMPC Formulation

The LMPC tries to compute a solution to the infinite time optimal control problem (6) by solving at time t of iteration j the finite time constrained optimal control problem

$$J^{\text{LMPC},j}_{t\to t+N}(x^j_t) = \min_{u_{t|t},\dots,u_{t+N-1|t}} \Bigg[\sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + Q^{j-1}(x_{t+N|t}) \Bigg] \tag{15a}$$

s.t.

$$x_{k+1|t} = f(x_{k|t}, u_{k|t}),\ \forall k \in [t,\cdots,t+N-1] \tag{15b}$$

$$x_{t|t} = x^j_t, \tag{15c}$$

$$x_{k|t} \in \mathcal{X},\ u_{k|k} \in \mathcal{U},\ \forall k \in [t,\cdots,t+N-1] \tag{15d}$$

$$x_{t+N|t} \in \ \mathcal{SS}^{j-1}, \tag{15e}$$

where (15b) and (15c) represent the system dynamics and initial condition, respectively. The state and input constraints are given by (15d). Finally (15e) forces the terminal state into the set $\mathcal{SS}^{j-1}$ defined in equation (11).

Let

$$\begin{aligned} \mathbf{u}^{*,j}_{t:t+N|t} &= [u^{*,j}_{t|t},\cdots,u^{*,j}_{t+N-1|t}] \\ \mathbf{x}^{*,j}_{t:t+N|t} &= [x^{*,j}_{t|t},\cdots,x^{*,j}_{t+N|t}] \end{aligned} \tag{16}$$

be the optimal solution of (15) at time t of the j-th iteration and $J^{\text{LMPC},j}_{0\to N}(x^j_t)$ 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)

$$u^j_t = u^{*,j}_{t|t}. \tag{17}$$

The finite time optimal control problem (15) is repeated at time $t+1$, based on the new state $x_{t+1|t+1} = x^j_{t+1}$ (15c), yielding a *moving* or *receding horizon* control strategy.

B. Repetitive Formulation

This section describes how we extended the work in [9] to deal with the autonomous racing problem as a repetitive task. Specifically, in [9] we assumed that the initial conditions are unchanged at each iteration (i.e. $x_S = x^j_0\ \forall j \geq 0$). On the other hand, for repetitive control the initial conditions of the j-th trial are function of the final state of the previous trial. This repetitive formulation allows us to design a controller which learns from data improving its performance with respect to problem (6), where x_0 is an optimization variable. In the specific autonomous racing application, given the track length s_F, we have

$$x^{j+1}_0 = x^j_{\bar{t}_j+1} - \theta,\ \forall j \geq 0 \tag{18}$$

with $\theta = [0,\ 0,\ 0,\ 0,\ 0,\ s_F]^T$ and

$$\bar{t}_j = \arg\min_t \{t+1 : x^j_{t+1} \in \mathcal{X}_F\} \tag{19}$$

being the time at which the transition between the the j-th and $j+1$-th iteration occurs. In the following we describe the properties of the proposed controller.

Remark 2: Usually in repetitive control the initial condition of the new iteration equals the final state of the previous one (i.e. $x^{j+1}_0 = x^j_{\bar{t}_j},\ \forall j \geq 0$). In (18) we decided to subtract θ to guarantee that at the current j-th iteration $s^j_t \in [0, s_F],\ \forall t \in [0, \bar{t}^j_t]$. Basically, we do not allow the curvilinear abscissa to increase.

Remark 3: Note that definition (18) implies that, $\forall x^j_{k+\bar{t}_j+1} \in \mathcal{X}_F$, we have

$$x^{j+1}_k = x^j_{k+\bar{t}_j+1} - \theta,\ \forall k \geq 0. \tag{20}$$

The above fact allows us to extend the sampled Safe Set beyond the finish line at s_F.

Assumption 2: We assume that $\mathcal{SS}^0$ is a given non-empty set and $\mathbf{x}^0 \in \mathcal{SS}^0$ is feasible and convergent to $\mathcal{X}_F$ with $x^1_0 = x^0_{\bar{t}_0} - \theta$.

The above *Assumption 2* is not restrictive in the autonomous racing application as one can always run a path following controller to track the lane centerline at a low speed.

Assumption 3: At time $t=0$ of the j-th iteration we assume that $J^{\text{LMPC},j}_{0\to N}(x^j_0) \leq J^{\text{LMPC},j}_{0\to N}(x^{j-1}_0),\ \forall j \geq 1$.

Assumption 3 implies that the initial condition of the j-th iteration does not penalize the controller performance at time $t=0$ of the j-th iteration. In general, this assumption is difficult to verify and it depends on the terminal invariant set $\mathcal{X}_F$ defined in (9). However, in the autonomous racing application we expect this assumption to be verified as at each iteration the controller crosses the finish line at a higher velocity, as confirmed by experimental results.

Theorem 1: Consider system (1) controlled by the LMPC (15) and (17). Let $\mathcal{SS}^j$ in (11) be the sampled safe set at the

j-th iteration. Let assumption 2 hold, then the LMPC (15) and (17) is feasible at $\forall j \geq 1$ and $t \in \mathbb{Z}_{+0}$. Moreover, the closed loop system (1) and (17) converges to the invariant set $\mathcal{X}_F$ at each j-th iteration $\forall j \geq 0$.

Proof: By *Assumption* 2 $\mathcal{SS}^0$ is a non-empty set. Furthermore combining assumption 2 and (18) we have that $x_0^0 = x_0^1$. Therefore at time $t = 0$ of the 1-st iteration

$$\begin{aligned} &[x_0^0, \cdots, x_N^0] \in \mathcal{SS}^{j-1} = \mathcal{SS}^0 \\ &[u_0^0, \cdots, u_{N-1}^0] \end{aligned} \tag{21}$$

satisfy input and state constraints (15b)-(15e) and therefore it is a feasible solution to the LMPC (15) and (17).

Assume that at time t of the j-th iteration the LMPC (15) and (17) is feasible and let $\mathbf{x}^{*,\mathbf{j}}_{\mathbf{t:t+N|t}}$ and $\mathbf{u}^{*,\mathbf{j}}_{\mathbf{t:t+N|t}}$ be the optimal trajectory and the input sequence as defined in (16). From (15c) and (17)

$$x_t^j = x_{t|t}^{*,j} \text{ and } u_t^j = u_{t|t}^{*,j}. \tag{22}$$

Moreover, as the state update (1) and (15b) are assumed to be identical we have $x_{t+1}^j = x_{t+1|t}^{*,j}$. Furthermore, the terminal constraint (15e) enforces $x_{t+N|t}^{*,j} \in \mathcal{SS}^{j-1}$ and, from

$$Q^{j-1}(x_{t+N|t}^{*,j}) = J_{t^* \to \infty}^{i^*}(x_{t+N|t}^{*,j}) = \sum_{k=t^*}^{\infty} h(x_k^{i^*}, u_k^{i^*}). \tag{23}$$

where $x_{t^*}^{i^*} = x_{t+N|t}^{*,j}$ with (i^*, t^*) being the minimizer in (13). At time $t+1$ of the j-th iteration the input sequence

$$[u_{t+1|t}^{*,j},\ u_{t+2|t}^{*,j},\ \ldots,\ u_{t+N-1|t}^{*,j},\ u_{t^*}^{i^*}], \tag{24}$$

and the related feasible state trajectory

$$[x_{t+1|t}^{*,j},\ x_{t+2|t}^{*,j},\ \ldots,\ x_{t+N-1|t}^{*,j},\ x_{t^*}^{i^*},\ x_{t^*+1}^{i^*}] \tag{25}$$

satisfy input and state constrains (15b)-(15c)-(15d). Therefore, (24)-(25) is a feasible solution for the LMPC (15) and (17) at time $t+1$.

Assume that at time $\bar{t}_j$ of the j-th iteration the LMPC (15) and (17) is feasible and let

$$\begin{aligned} \mathbf{u}^{*,j}_{\bar{t}_j:\bar{t}_j+N|\bar{t}_j} &= [u^{*,j}_{\bar{t}_j|\bar{t}_j}, \cdots, u^{*,j}_{\bar{t}_j+N-1|\bar{t}_j}] \\ \mathbf{x}^{*,j}_{\bar{t}_j:\bar{t}_j+N|\bar{t}_j} &= [x^{*,j}_{\bar{t}_j|\bar{t}_j}, \cdots, x^{*,j}_{\bar{t}_j+N|\bar{t}_j}] \end{aligned} \tag{26}$$

be the optimal solution. By definition (18) at time $\bar{t}_j + 1$ of the j-th iteration the system reaches the terminal invariant set and the following iteration $j+1$ starts. In the following we construct a feasible solution to the LMPC (15) and (17) at $t = 0$ of the j+1, using (26). Firstly we notice that as the state update in (1) and (15b) are assumed identical $x_{\bar{t}_j+1}^j = x^{*,j}_{\bar{t}_j+1|\bar{t}_j}$ and by definitions (18)-(20), we have

$$x_0^{j+1} = x_{\bar{t}_j+1}^j - \theta = x^{*,j}_{\bar{t}_j+1|\bar{t}_j} - \theta. \tag{27}$$

By the definition of $\bar{t}_j$ we have $x^{*,j}_{\bar{t}_j+N|\bar{t}_j} \in \mathcal{X}_F \subset \mathcal{SS}^{j-1}$ and therefore from (20) it exists $k \geq 0$ such that

$$x^{*,j}_{\bar{t}_j+N|\bar{t}_j} = x_{t^*}^{i^*} = x_{k+\bar{t}_j+1}^{i^*} \tag{28}$$

where (i^*, t^*) are the minimizer in (13) with $i^* < j$. Moreover we notice that

$$x^{*,j}_{\bar{t}_j+N|\bar{t}_j} - \theta = x^{i^*}_{k+\bar{t}_j+1} - \theta = x_k^{i^*+1} \in \mathcal{SS}^{i^*+1} \subset \mathcal{SS}^j. \tag{29}$$

Furthermore as $x_{k+1}^{i^*+1} \in \mathcal{SS}^j$, the trajectory

$$[x^{*,j}_{\bar{t}_j+1|\bar{t}_j} - \theta, \cdots, x^{*,j}_{\bar{t}_j+N|\bar{t}_j} - \theta = x_k^{i^*+1}, x_{k+1}^{i^*+1}] \tag{30}$$

is a feasible trajectory for the LMPC (15) and (17) at time t of iteration $j+1$.

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

At each j-th iteration the controller tries to steer the system to the invariant set $\mathcal{X}_F$ and each iteration can be analyzed independently from the others. Therefore the convergence of the closed loop system (1) and (17) to the invariant set $\mathcal{X}_F$ follows from *Theorem 1* in [13]. ■

Theorem 2: Consider system (1) in closed loop with the LMPC controller (15) and (17). Let $\mathcal{SS}^j$ be the sampled safe set at the j-th iteration as defined in (11). Let assumptions 2-3 hold, then the iteration cost $J^j_{0\to\infty}(\cdot)$ does not increase with the iteration index j.

Proof: First, we find a lower bound on the j-th iteration cost $J^j_{0\to\infty}(\cdot),\ \forall\ j > 0$. Consider the realized state and input sequence (10) at the j-th iteration, which collects the first element of the optimal state and input sequence to the LMPC (15) and (17) at time $t,\ \forall t \in \mathbb{Z}_{0+}$. By the definition of the iteration cost in (12) and from assumption 3, we have

$$\begin{aligned} J^{j-1}_{0\to\infty}(x_S) &= \sum_{t=0}^{\infty} h(x_t^{j-1}, u_t^{j-1}) = \\ &= \sum_{t=0}^{N-1} h(x_t^{j-1}, u_t^{j-1}) + \sum_{t=N}^{\infty} h(x_t^{j-1}, u_t^{j-1}) \geq \\ &\geq \sum_{t=0}^{N-1} h(x_t^{j-1}, u_t^{j-1}) + Q^{j-1}(x_N^{j-1}) \geq \\ &\geq \min_{u_0,\ldots,u_{N-1}} \left[\sum_{k=0}^{N-1} h(x_k, u_k) + Q^{j-1}(x_N) \right] = \\ &= J^{\text{LMPC},j}_{0\to N}(x_0^{j-1}) \geq J^{\text{LMPC},j}_{0\to N}(x_0^j). \end{aligned} \tag{31}$$

Given the above expression the proof follows from *Theorem 2* in [13]. ■

V. Experimental setup and results

We implemented the proposed control strategy on a 1/10-scale open source vehicle platform called the Berkeley Autonomous Race Car (BARC)[1]. A video of the experiment can be found at `https://automatedcars.space/home/2016/12/22/learning-mpc-for-autonomous-racing`.

Since the sampled safe set $\mathcal{SS}^{j-1}$ is a set of discrete points, the LMPC (15), (17) is a Mixed Integer Programming (MIP) problem and its implementation is prohibitive in real time. To achieve real time feasibility we relax the Safe Set $\mathcal{SS}^{j-1}$ to its convex hull and $Q^{j-1}(\cdot)$ to the barycentric approximation. The details of the LMPC relaxation are not discussed here for space constraints and we refer to [9] for further details.

The relaxed LMPC is solved at a sample rate of $10\,\mathrm{Hz}$ and optimal commands are sent to the motor and steering servo at the same rate. The code is implemented in Python and Julia [14] using ROS (Robot Operating System). The LMPC is written as an optimization problem and it is solved using Ipopt [15].

In order to find a feasible trajectory to initialize the LMPC (15) and (17), we implemented a path following controller that tracks the centerline of the road at the constant speed of $1\,\mathrm{m\,s^{-1}}$. This feasible trajectory is used to compute $\mathcal{SS}^0$, $Q^0(\cdot)$ and their approximations [9].

We tested the proposed controller on a race track of $19\,\mathrm{m}$ and with maximum curvatures of $1.2\,\mathrm{m}^{-1}$ shown in Figure 3. The maximum curvature of the track matches the maximal curvature our test system is able to drive. Note that a vertical black line marks the starting line of the track in Figures 4 and 3. After 12 iterations the relaxed LMPC converges to a steady state trajectory. Figure 1 illustrates that the lap time decreases from $22\,\mathrm{s}$ to a steady state value of $7.55\,\mathrm{s}\pm 0.25\,\mathrm{s}$.

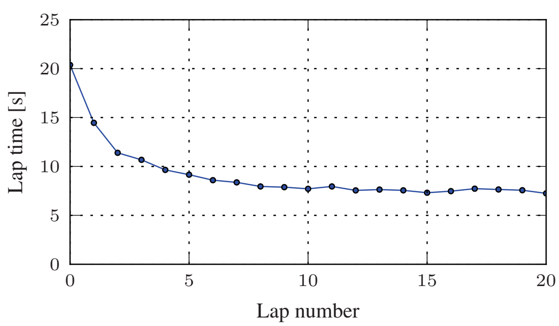

Fig. 1. Decreasing lap time

Figure 4 shows the closed loop trajectory at different iterations. We notice that in the 1st learning iteration the controller deviates from the track centerline to cut the curves. However, the controller must satisfy the terminal constraints (15e) which force the predicted final state to lie in $\mathcal{SS}^{j-1}$ and prevent it from performing a higher velocity lap. After 8 iterations, the controller learned to drive the vehicle at higher velocities. This more aggressive behavior is the result of a bigger sampled Safe Set learned from previous iteration data. A projection of the sampled sampled Safe Set $\mathcal{SS}^{j-1}$

[1]More information at the project site barc-project.com

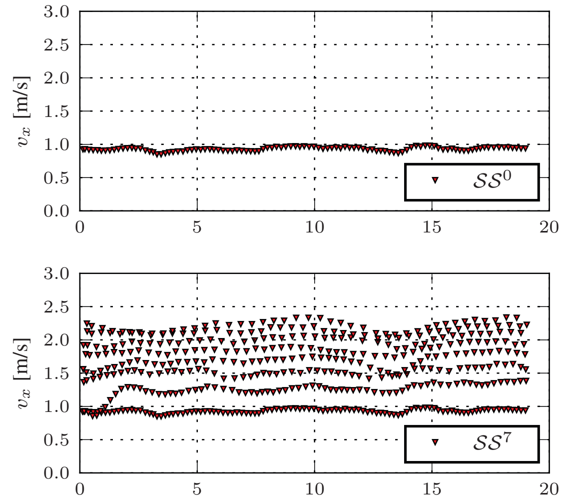

Fig. 2. Projection of $\mathcal{SS}^{j-1}$ into the s-v_x plane at the j-th iteration, for $j = 1$ and $j = 8$.

onto the s-v_x plane for the 1st and 8th iteration is shown in Figure 2.

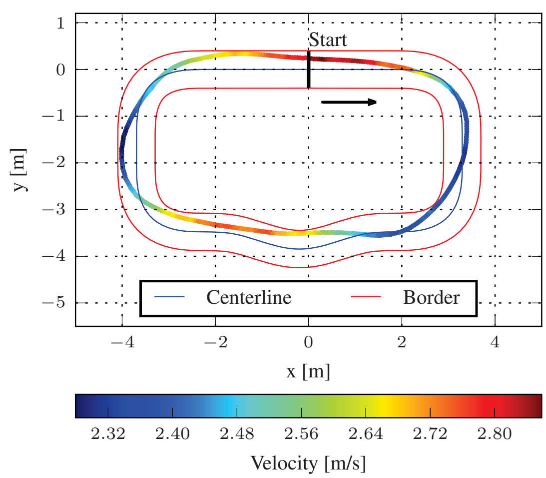

Fig. 3. Velocity over x,y

Furthermore, Figure 4 presents the steady state trajectory. We noticed that the first and final point of the steady state trajectory coincide. On the other hand, the first and final point of the 1st and 8th iteration do not coincide. This shows the need of the proposed repetitive LMPC, which allows to continuously run the vehicle on the track until the controller finds the best trajectory to cross the finish line (i.e. $x^{\infty}_{\bar{t}_\infty+1} = x^{\infty}_0 \neq x^0_0$). We underline that this behavior cannot be accomplished with the iterative controller presented in [9], as the controller starts at the same initial state in each iteration (i.e. $x^0_0 = x^{\infty}_0,\ \forall j \geq 0$).

Figure 3 shows the steady state trajectory and its velocity profile using a color scale tailored on the minimum and maximum steady state velocity. The controller correctly

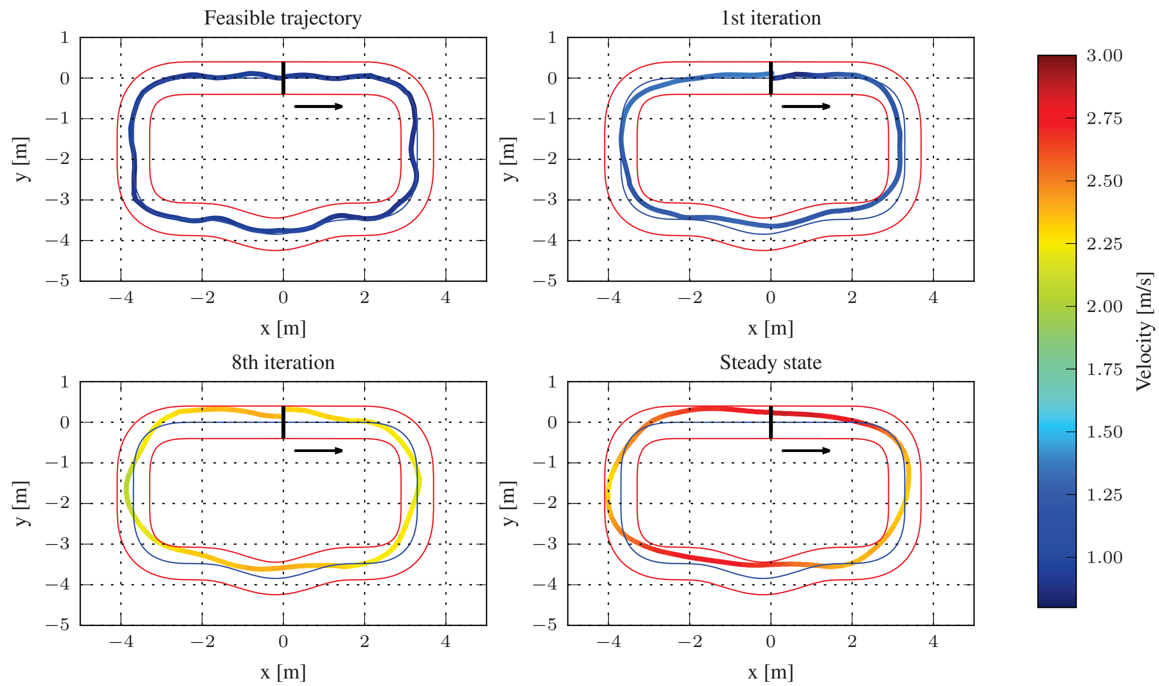

Fig. 4. Evolution of the closed loop trajectory over the iterations

understands the benefit of cutting the curves saturating the road constraints. Furthermore, we notice that the controller decelerate until the midpoint of the curve and accelerate afterwards. This behavior has been shown to be optimal for race driving [8].

VI. Conclusions

In this paper, a learning model predictive control for the racing problem that exploits information from the previous laps to improve the performance of the closed loop system over iterations is presented. A sampled safe set and a terminal cost learned from data are used to guaranteed the recursive stability and performance improvement of the control framework. Furthermore, we proposed a repetitive formulation that allows the controller to run continuously on the race track. We tested the proposed control strategy on a 1:10 scale experimental setup to prove real time feasibility of the proposed controller. Finally, experimental results confirmed that the control is able to improve its performance until it converges to a steady state trajectory.

References

[1] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *11th International Symposium on Advanced Vehicle Control*, 2012.

[2] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 4136–4141.

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

[4] A. Alleyne, "A comparison of alternative intervention strategies for unintended roadway departure (urd) control," *Vehicle System Dynamics*, vol. 27, no. 3, pp. 157–186, 1997.

[5] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[6] R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear mpc in real-time," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 2505–2510.

[7] N. R. Kapania and J. C. Gerdes, "Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control," in *American Control Conference (ACC), 2015*. IEEE, 2015.

[8] P. A. Theodosis and J. C. Gerdes, "Generating a racing line for an autonomous racecar using professional driving techniques," in *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*. American Society of Mechanical Engineers, 2011, pp. 853–860.

[9] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *American Control Conference (ACC), 2017*.

[10] V. Cerone, D. Piga, and D. Regruto, "Set-membership lpv model identification of vehicle lateral dynamics," *Automatica*, vol. 47, no. 8, pp. 1794–1799, 2011.

[11] 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.

[12] 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)*.

[13] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework." *IEEE Transaction on Automatic Control*, 2017.

[14] J. Bezanzon, S. Karpinski, V. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," in *Lang.NEXT*, Apr. 2012. [Online]. Available: http://julialang.org/images/lang.next.pdf

[15] 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.