Autonomous Racing using Learning Model Predictive Control

Ugo Rosolia, Ashwin Carvalho and Francesco Borrelli

***Abstract*— A novel learning Model Predictive Control technique is applied to the autonomous racing problem. The goal of the controller is to minimize the time to complete a lap. The proposed control strategy uses the data from previous laps to improve its performance while satisfying safety requirements. A system identification technique is proposed to estimate the vehicle dynamics. Simulation results with the high fidelity simulator software CarSim show the effectiveness of the proposed control scheme.**

I. INTRODUCTION

Nonlinear model predictive control is an appealing technique for autonomous driving because of its ability to handle input and state constraints as well as nonlinearities introduced by the vehicle dynamics. Recently, the effectiveness and the real-time feasibility of this control strategy has been demonstrated in [1], [2]. However, real-time path generation for racing applications in real-world conditions remains challenging. If the objective is the minimization of the lap time, the controller has to plan the trajectory on a sufficiently long time interval to avoid aggressive maneuvers that would lead the vehicle outside the track. For instance, in a straight line before a curve the controller should limit the velocity to allow turning. Furthermore, when solving the problem with a receding horizon approach, the resulting closed-loop trajectory is not guaranteed to be optimal for the original problem.

In [3] two approaches are presented for autonomous racing. In the first one, a high level MPC controller is used to generate a feasible trajectory for the low level tracking MPC. In the second approach, a single layer Model Predictive Contouring Control (MPCC) is presented, where the controller objective is a trade-off between the progress along the track and the contouring error. Also in [4], the authors compared two approaches, the first one based on a tracking MPC, and, the second one, based on a MPC that aims to minimize the lap time. The authors pointed out the importance of the horizon length, which is necessary to reach good performance. In [5] the authors proposed an iterative learning control (ILC) approach for autonomous racing. The proposed ILC tracks an aggressive trajectory computed off-line using the techniques proposed in [6]. The authors showed the effectiveness of the proposed ILC with experimental testing on a full size vehicle.

In this paper we propose to tackle the minimum lap time problem as reference free iterative control problem using the novel technique presented in [7]. At each j-th iteration the controller starts from the same starting point and it has to minimize the traveling time to cross the finish line. Under no model mismatch, the Learning MPC (LMPC) presented in [7] guaranties that *(i):* the j-th iteration's 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 iteration j if they were satisfied at iteration $j-1$ (recursive feasibility), *(iii):* if the controller goal is to regulate the system to an equilibrium point x_F, then such an equilibrium point is asymptotically stable, *(iv)*: if system converges to a steady state trajectory as the number of iterations j goes to infinity, then such a trajectory is locally optimal. The assumption of no model mismatch is not satisfied in the racing application especially due to the highly dynamic range of maneuvers. Therefore, in this work we propose to couple the LMPC with a system identification algorithm and we show, through high fidelity simulations with CarSim, that the controller successfully improves the overall objective function and it converges to a steady state solution. Furthermore, we propose a problem relaxation that reduces the computational burden associated with [7] and satisfies real-time requirements.

As mentioned, we study also the case of model parameters uncertainty and their iterative estimation. In general, the MPC design under the presence of model uncertainty may be challenging. In [8] the author proposes to use two different models, a nominal one used to check the constraint satisfaction and a learnt one to improve performance. However, for the racing problem it may be difficult to define a priori a model that guaranties the feasibility. A different approach is proposed in [9], where MPC algorithm uses a simple a priori vehicle model and a learned disturbance model. The authors proved the effectiveness of this control strategy for a path following problem. In the racing problem, we would like to have a model of vehicle when operating close to its handling capability. Unfortunately, modelling the vehicle lateral dynamics is challenging. Also when the linear model assumption holds, the tire stiffness are speed and environment dependent. In [10] the lateral dynamics are estimated using a LPV system and a set membership algorithm has been successfully implemented. This procedure might be computationally demanding for on-line model learning. In this paper we propose an identification approach which exploits information from the previous iterations. At each time step t of the MPC algorithm, system dynamics are estimated using only input-output data from previous iterations close to the current system state $x(t)$.

This paper is organized as follows: in Section II we intro-

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

duce the vehicle model and we formalize the minimum time problem. The iterative formulation is illustrated in Section III, where we introduce the quantities necessary to implement the LMPC. In Section IV, the system identification procedure is described. The LMPC and its relaxation are illustrated in Section V. Finally, in Section VI we test the proposed control logic on an a section of a race track. Section VII provides final remarks.

II. PROBLEM DEFINITION

A. Vehicle Model

The vehicle dynamics is described by,

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

where $f(x_t, u_t)$ is the vehicle dynamic state update equation. The vectors x_t and u_t in (1) collect the states and inputs of the vehicle at time t,

$$x_t = [v_{x_t}\ v_{y_t}\ \dot{\psi}_t\ e_{\psi_t}\ e_{y_t}\ s_t] \tag{2a}$$
$$u_t = [a_t\ \delta_t], \tag{2b}$$

where s_t represents the distance travelled along the center-line of the road, e_{y_t} and e_{ψ_t} the lateral distance and heading angle error between the vehicle and the path. v_{x_t}, v_{y_t} and $\dot{\psi}_t$ are the vehicle longitudinal velocity, lateral velocity and yaw rate, respectively. The inputs are the longitudinal acceleration a_t and the steering angle δ_t. For more details on the curvilinear abscissa reference frame we refer to [11].

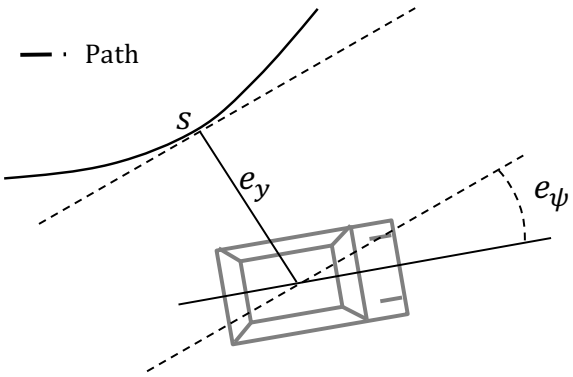

Fig. 1. Representation of the vehicle in the curvilinear abscissa reference frame

B. Minimum Time Problem

The goal of the controller is to minimize the time to cross the finish line at s_{target}. More formally, the goal of the controller is to solve the following constrained infinite horizon optimal control problem

$$J^*_{0\to\infty}(x_S) = \min_{u_0, u_1, \dots} \sum_{k=0}^{\infty} h(x_k, u_k) \tag{3a}$$
$$\text{s.t. } x_{k+1} = f(x_k, u_k),\ \forall k \geq 0 \tag{3b}$$
$$x_0 = x_S, \tag{3c}$$
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \geq 0 \tag{3d}$$

where equations (3b) and (3c) represent the vehicle dynamics and the initial condition, and (3d) are the state and input constraints. Note that In our problem the state constraint $\mathcal{X}$ is a convex set representing the road boundaries. The stage cost, $h(\cdot,\cdot)$, in equation (3a) is defined as

$$h(x_k, u_k) = \begin{cases} 1 & \text{if } x_k \notin \mathcal{L} \\ 0 & \text{if } x_k \in \mathcal{L} \end{cases}, \tag{4}$$

where $\mathcal{L}$ is the set of points beyond the finish line at s_{target},

$$\mathcal{L} = \left\{ x \in \mathbb{R}^6 :\ x e_6^T = s > s_{target} \right\} \tag{5}$$

where e_6 is the the 6-th standard basis in $\mathbb{R}^6$.

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 after the finish line. Namely, we assume that $\mathcal{L}$ is a control invariant set, thus

$$\forall x_k \in \mathcal{L}, \exists u_k \in \mathcal{U} : x_{k+1} = f(x_k, u_k) \in \mathcal{L}. \tag{6}$$

III. LMPC PRELIMINARIES

The constrained infinite horizon optimal control problem in (3) is difficult to solve, especially in real time. Therefore, we implemented the Learning MPC for iterative tasks presented in [7], where at each iteration the controller computes a trajectory that brings the system from the staring point x_S to the invariant set $\mathcal{L}$. The data from each iteration are used to improve the performance of the controller until the controller converges to a local optimal solution, we refer to [7] for further details. In the following we introduce the notation that will be used to implement the Learning MPC for the racing application.

A. Sampled Safe Set

At the j-th iteration the vectors

$$\mathbf{u}^j = [u_0^j,\ u_1^j,\ \dots,\ u_t^j,\ \dots], \tag{7a}$$
$$\mathbf{x}^j = [x_0^j,\ x_1^j,\ \dots,\ x_t^j,\ \dots], \tag{7b}$$

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,

$$x_t^j = [v_{x_t}^j\ v_{y_t}^j\ \dot{\psi}_t^j\ e_{\psi_t}^j\ e_{y_t}^j\ s_t^j] \tag{8a}$$
$$u_t^j = [a_t^j\ \delta_t^j]. \tag{8b}$$

Remark 1: In (7b) we have that $x_0^j = x_S,\ \forall j \geq 0$, as the vehicle starts from the same initial point x_S at each j-th iteration.

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

$$\mathcal{SS}^j = \left\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{\infty} x_t^i \right\}. \tag{9}$$

$\mathcal{SS}^j$ is the collection of all state trajectories at iteration i for $i \in M^j$. M^j in equation (9) is the set of indexes k associated with successful iterations k for $k \leq j$, defined as:

$$M^j = \left\{ k \in [0, j] : \lim_{t\to\infty} x_t^k = \mathcal{L} \right\}. \tag{10}$$

B. Iteration Cost

At time t of the j-th iteration the cost-to-go associated with the closed loop trajectory (7b) and input sequence (7a) is defined as

$$J^j_{t\to\infty}(x^j_t) = \sum_{k=t}^{\infty} h(x^j_k, u^j_k), \tag{11}$$

where $h(\cdot,\cdot)$ is the stage cost of the problem (3). We define the j-th iteration cost as the cost (11) of the j-th trajectory at time $t = 0$,

$$J^j_{0\to\infty}(x^j_0) = \sum_{k=0}^{\infty} h(x^j_k, u^j_k). \tag{12}$$

$J^j_{0\to\infty}(x^j_0)$ quantifies the controller performance at each j-th iteration.

Remark 2: In equations (11)-(12), x^j_k and u^j_k are the realized state and input at the j-th iteration, as defined in (7).

Remark 3: At each j-th iteration the system is initialized at the same starting point $x^j_0 = x_S$; thus we have $J^j_{0\to\infty}(x^j_0) = J^j_{0\to\infty}(x_S)$.

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

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

Remark 4: The function $Q^j(\cdot)$ in (13) assigns to every point in the sampled safe set, $\mathcal{SS}^j$, the minimum cost-to-go along the trajectories in $\mathcal{SS}^j$.

IV. SYSTEM IDENTIFICATION

In this section the system identification technique is described. At the j-th iteration, the trajectories in the sampled safe set, $\mathcal{SS}^{j-1}$, and the data from the current iteration are used to estimate the system dynamics. In particular we assume that the system dynamic update at time t of the j-th iteration is given by

$$x^j_{t+1} = f^j_t(x^j_t, u^j_t) = \bar{g}(x^j_t, u^j_t) + g^j_t(x^j_t, u^j_t) \tag{15}$$

where $\bar{g}(x^j_t, u^j_t) \in \mathbb{R}^n$ is known function that represents the known dynamic of the system, and at time t of the j-th iteration the function

$$g^j_t(x^j_t, u^j_t) = [g^j_{1,t}(\cdot), \cdots, g^j_{n,t}(\cdot)]^T \in \mathbb{R}^n \tag{16}$$

is identified using a least mean square technique. The entries of $g^j_t(\cdot)$ are defined as

$$g^j_{i,t}(\cdot) = \gamma^j_{i,t}\theta^j_{i,t},\ i \in \{1, \cdots, n\} \tag{17}$$

where $\gamma^j_{i,t},\ i \in \{1, \cdots, n\}$ is the feature vector that may change for different states and $\theta^j_{i,t},\ \forall i \in \{1, \cdots, n\}$ is the parameter vector which is estimated on-line.

Remark 5: We refer to the Appendix I for further details on how we implemented (15)-(17) for the racing problem.

Furthermore, in order to estimate the parameter vector $\theta^j_{i,t} \in \mathbb{R}^{p_i},\ \forall i \in \{1, \cdots, n\}$, we implemented the following least mean square technique,

$$\theta^j_{i,t} = \underset{\bar{\theta}}{\operatorname{argmin}} ||\phi^j_{i,t}\theta - y^j_{i,t}||_2,\ \forall i \in \{1, \cdots, n\}, \tag{18}$$

where the vector $y^j_{t,i}$ selects a subset of the collected data using a distance-based criterion. In particular, at time t of the j-th iteration we define the time index $l^{t,j}_k$ with $k \leq j$, for which $x^k_{l^{t,j}_k}$ is the closest point to the current system state x^j_t,

$$l^{t,j}_k = \underset{\bar{l}}{\operatorname{argmin}} ||x^j_t - x^k_{\bar{l}}||_2. \tag{19}$$

Afterwards, the measurement vector, $y^j_{t,i}$, in (18) is constructed using the data collected in the last $\bar{n}$ steps, and, for the last n_j+1 trajectories in $\mathcal{SS}^{j-1}$, the data collected $\bar{n}$ steps before and n steps after $l^{t,j}_k$,

$$y^j_{i,t} = \begin{bmatrix} \Big(x^j_{l^{t,j}_j} - \bar{g}(x_{l^{t,j}_j\text{-}1}, u_{l^{t,j}_j\text{-}1})\Big)e^T_i \\ \vdots \\ \Big(x^j_{l^{t,j}_j\text{-}\bar{n}} - \bar{g}(x^j_{l^{t,j}_j\text{-}\bar{n}\text{-}1}, u^j_{l^{t,j}_j\text{-}\bar{n}\text{-}1})\Big)e^T_i \\ \Big(x^{j\text{-}1}_{l^{t,j}_{j\text{-}1}+n} - \bar{g}(x^{j\text{-}1}_{l^{t,j}_{j\text{-}1}+n\text{-}1}, u^{j\text{-}1}_{l^{t,j}_{j\text{-}1}+n\text{-}1})\Big)e^T_i \\ \vdots \\ \Big(x^{j\text{-}1}_{l^{t,j}_{j\text{-}1}\text{-}\bar{n}} - \bar{g}(x^{j\text{-}1}_{l^{t,j}_{j\text{-}1}\text{-}\bar{n}\text{-}1}, u^{j\text{-}1}_{l^{t,j}_{j\text{-}1}\text{-}\bar{n}\text{-}1})\Big)e^T_i \\ \vdots \\ \Big(x^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}+n} - \bar{g}(x^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}+n\text{-}1}, u^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}+n\text{-}1})\Big)e^T_i \\ \vdots \\ \Big(x^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}\text{-}\bar{n}}e^T_i - \bar{g}(x^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}\text{-}\bar{n}\text{-}1}, u^{j\text{-}n_j}_{l^{t,j}_{j\text{-}n_j}\text{-}\bar{n}\text{-}1})\Big)e^T_i \end{bmatrix} \in \mathbb{R}^{\bar{N}}, \tag{20}$$

$i \in \{1, \cdots, n\}$, where e_i is the standard basis in $\mathbb{R}^n$, and $\bar{N} = \bar{n}$+1+$n_j(\bar{n}$+n+1) is the number of points used for the system identification.

Remark 6: $\bar{n}, n_j, n$ are tuning variables which are chosen by the designer.

In (18), the regressor matrix $\phi^j_{t,i}, \forall i \in [v_x, v_y, \dot{\psi}]$ is computed using the feature vectors, and the data collected in

$\mathcal{SS}^{j-1}$ and current iterations,

$$\phi^j_{i,t} = \begin{bmatrix} \gamma^{j}_{i_{l^{t,j}_{j}\text{-}1}} \\ \vdots \\ \gamma^{j}_{i_{l^{t,j}_{j}\text{-}\bar{n}\text{-}1}} \\ \gamma^{j\text{-}1}_{i_{l^{t,j}_{j\text{-}1}+n\text{-}1}} \\ \vdots \\ \gamma^{j\text{-}1}_{i_{l^{t,j}_{j\text{-}1}\text{-}\bar{n}\text{-}1}} \\ \vdots \\ \gamma^{j\text{-}n_j}_{i_{l^{t,j}_{j\text{-}n_j}+n\text{-}1}} \\ \vdots \\ \gamma^{j\text{-}n_j}_{i_{l^{t,j}_{j\text{-}n_j}}\text{-}\bar{n}\text{-}1} \end{bmatrix} \in \mathbb{R}^{\bar{N}\times p_i},\ i \in \{1,\cdots,n\} \tag{21}$$

V. LMPC CONTROL DESIGN

A. LMPC Formulation

The LMPC tries to compute a solution to the infinite time optimal control problem (3) by solving at time t of iteration j the finite time constrained optimal control problem

$$J^{\text{LMPC},j}_{t\to t+N}(x^j_t) = \min_{u_{t|t},\ldots,u_{t+N-1|t}} \Bigg[\sum_{k=t}^{t+N-1} h(x_{k|t},u_{k|t}) + \\ + Q^{j-1}(x_{t+N|t}) \Bigg] \tag{22a}$$

s.t.

$$x_{k+1|t} = f^j_t(x_{k|t},u_{k|t}),\ \forall k \in [t,\cdots,t+N-1] \tag{22b}$$
$$x_{t|t} = x^j_t, \tag{22c}$$
$$x_{k|t} \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [t,\cdots,t+N-1] \tag{22d}$$
$$x_{t+N|t} \in\ \mathcal{SS}^{j-1}, \tag{22e}$$

where (22b) and (22c) represent the system dynamics in (15) and initial condition, respectively. The state and input constraints are given by (22d). Finally (22e) forces the terminal state into the set $\mathcal{SS}^{j-1}$ defined in equation (9). Let

$$\begin{aligned} \mathbf{u}^{*,j}_{t:t+N|t} &= [u^{*,j}_{t|t},\cdots,u^{*,j}_{t+N-1|t}] \\ \mathbf{x}^{*,j}_{t:t+N|t} &= [x^{*,j}_{t|t},\cdots,x^{*,j}_{t+N|t}] \end{aligned} \tag{23}$$

be the optimal solution of (22) 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)

$$u^j_t = u^{*,j}_{t|t}. \tag{24}$$

The finite time optimal control problem (22) is repeated at time $t+1$, based on the new state $x_{t+1|t+1} = x^j_{t+1}$ (22c), yielding a *moving* or *receding horizon* control strategy.

B. LMPC Relaxation

As the sampled safe set in (9) is a set of discrete points, the optimal control problem in (22) is a Nonlinear Mixed Integer Programming, therefore the LMPC (22) and (24) may be computationally challenging to solve in real-time. Therefore, we introduce the time varying approximated safe set, $\tilde{\mathcal{SS}}^{j-1}_t$, and the time varying approximated $Q^{j-1}(\cdot)$ function, $\tilde{Q}^{j-1}_t(\cdot)$. At time t of the j-th iteration, we approximate $\mathcal{SS}^{j-1}$ with the time varying approximated sampled safe set,

$$\tilde{\mathcal{SS}}^{j-1}_t = \Big\{ x \in \mathbb{R}^6, \lambda \in [0,1] : \\ x \in \lambda \tilde{\mathbf{x}}^{j-1}_t + (1-\lambda)\tilde{\mathbf{x}}^{j-2}_t \Big\} \tag{25}$$

where $\tilde{\mathbf{x}}^{j-1}_t$ approximates locally the j-1-th trajectory, $\mathbf{x}^{j-1}$, using a 5-th order polynomial function,

$$\tilde{\mathbf{x}}^{j-1}_t = \Big\{ x \in \mathbb{R}^6 :\ \forall i \in \{v_x,\ v_y,\ \dot{\psi},\ e_\psi,\ e_y\}, \\ i = \ [s^5\ s^4\ s^3\ s^2\ s\ 1]\ \Gamma^{j-1}_{t,i} \Big\}, \tag{26}$$

being $\Gamma^{j-1}_{t,i}$ the solution to the following least mean square problem

$$\Gamma^{j-1}_{t,i} = \underset{\Gamma}{\text{argmin}} \Bigg|\Bigg| \begin{bmatrix} i^{j-1}_{l^{t,j}_{j-1}+4N} \\ \vdots \\ i^{j-1}_{l^{t,j}_{j-1}} \end{bmatrix} + \\ - \begin{bmatrix} s^5_{l^{t,j}_{j\text{-}1}+4\text{N}} & \cdots & s_{l^{t,j}_{j\text{-}1}+4\text{N}} & 1 \\ \vdots & & \vdots & \\ s^5_{l^{t,j}_{j\text{-}1}} & \cdots & s_{l^{t,j}_{j\text{-}1}} & 1 \end{bmatrix} \Gamma \Bigg|\Bigg|_2, \tag{27}$$

where $l^{t,j}_{j\text{-}1}$ is the time index defined in equation (19).

Remark 7: Note that $\tilde{\mathbf{x}}^{j-2}_t$ is defined replacing j-1 with j-2 in the above definition.

Furthermore, we introduce the time varying function $C^{j-1}_t(\cdot)$, which at time t of the j-th iteration approximates the cost to go along the j-1-th trajectory,

$$C^{j-1}_t(x) = \begin{cases} [s^5\ s^4\ s^3\ s^2\ s\ 1]\Delta^{j-1}_t, & \text{if } x \in \tilde{\mathbf{x}}^{j-1}_t \\ +\infty, & \text{if } x \notin \tilde{\mathbf{x}}^{j-1}_t \end{cases}, \tag{28}$$

where $\Delta_{t,i}^{j-1}$ is the solution to the following least mean square problem

$$\Delta_t^{j-1} = \underset{\Delta}{\operatorname{argmin}} || \begin{bmatrix} J_{t\to\infty}^{j-1}(x_{l_{j-1}^{t,j}+4\mathrm{N}}^{j-1}) \\ \vdots \\ J_{t\to\infty}^{j-1}(x_{l_{j-1}^{t,j}}^{j-1}) \end{bmatrix} + - \begin{bmatrix} s_{l_{j-1}^{t,j}+4\mathrm{N}}^{5} & \cdots & s_{l_{j-1}^{t,j}+4\mathrm{N}} & 1 \\ \vdots & & \vdots & \\ s_{l_{j-1}^{t,j}}^{5} & \cdots & s_{l_{j-1}^{t,j}} & 1 \end{bmatrix} \Delta ||_2. \tag{29}$$

Remark 8: Note that $C_t^{j-2}(\cdot)$ is defined replacing j-1 with j-2 in the above definition.

The $C_t^{j-1}(\cdot)$ function in (28) is used to define the continuous time varying approximation of $Q^{j-1}(\cdot)$,

$$\tilde{Q}_t^{j-1}(x,\lambda) = \begin{cases} \lambda C_t^{j-1}(x) + (1-\lambda) C_t^{j-2}(x), & \\ \qquad\qquad \text{if } (x,\lambda) \in \tilde{\mathcal{SS}}_t^{j-1} & \\ +\infty, \qquad \text{if } (x,\lambda) \in \tilde{\mathcal{SS}}_t^{j-1} & \end{cases}. \tag{30}$$

Finally, we reformulate the LMPC in (22) and (24) using the time varying approximation of $\mathcal{SS}^{j-1}$ and of the $Q^{j-1}(\cdot)$ function, to have a computationally tractable problem. We define the following constrained finite time optimal control problem,

$$\tilde{J}_{t\to t+N}^{\text{LMPC},j}(x_t^j) = \min_{\lambda, u_{t|t},\ldots,u_{t+N-1|t}} \Big[\sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + + \tilde{Q}_t^{j-1}(x_{t+N|t}, \lambda) \Big] \tag{31a}$$

s.t.

$$x_{k+1|t} = f_t^j(x_{k|t}, u_{k|t}),\ \forall k \in [t, \cdots, t+N-1] \tag{31b}$$

$$x_{t|t} = x_t^j, \tag{31c}$$

$$x_{k|t} \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \in [t, \cdots, t+N-1] \tag{31d}$$

$$(x_{t+N|t}, \lambda) \in \tilde{\mathcal{SS}}_t^{j-1}. \tag{31e}$$

Let

$$\begin{aligned} \tilde{\mathbf{u}}_{t:t+N|t}^{*,j} &= [\tilde{u}_{t|t}^{*,j}, \cdots, \tilde{u}_{t+N-1|t}^{*,j}] \\ \tilde{\mathbf{x}}_{t:t+N|t}^{*,j} &= [\tilde{x}_{t|t}^{*,j}, \cdots, \tilde{x}_{t+N|t}^{*,j}] \end{aligned} \tag{32}$$

be the optimal solution of (31) at time t of the j-th iteration and $\tilde{J}_{0\to N}^{\text{LMPC},j}(x_t^j)$ the corresponding optimal cost. Then, at time t of the iteration j, the first element of $\tilde{\mathbf{u}}_{t:t+N|t}^{*,j}$ is applied to the system (1)

$$u_t^j = \tilde{u}_{t|t}^{*,j}. \tag{33}$$

The finite time optimal control problem (31) is repeated at time $t+1$, based on the new state $x_{t+1|t+1} = x_{t+1}^j$ (31c).

VI. SIMULATION RESULTS

Simulations are performed in CarSim which uses a high-fidelity vehicle model to simulate the vehicle dynamics. The parameters of the CarSim vehicle and tire models were identified using data collected from our experimental passenger vehicle. The nonlinear optimization problem (31) is solved using NPSOL [12]. To verify the real-time feasibility of the LMPC strategy, we successfully ran the closed-loop simulation on a dSpace MicroAutobox II embedded computer (900MHz IBM PowerPC processor) at a sampling time of 100ms. The LMPC is discretized at 100ms and the horizon $N = 10$. The system identifiation algorithms is implemented using $n = \bar{n} = 50$ and $n_j = 2$. Future work will focus on the implementation on our experimental vehicle.

In order to implement the LMPC (31) and (33), we computed a feasible trajectory, $\mathbf{x^0}$, that drives the from x_S to the invariant set $\mathcal{L}$ using a path following controller at a low velocity. For more details on the path following controller, we refer to [13]. The first feasible trajectory, $\mathbf{x^0}$, is used to construct $\tilde{\mathcal{SS}}_t^0$ and $\tilde{Q}_t^0$, needed to initialize the first iteration of the LMPC (31) and (33). Parameters used in the controller are reported in Table I.

To effectively illustrate the results, the controller is tested on a single corner of a race track. The track centerline, starting position and finish line are shown in Figure 2. In particular, we have a straight in the first section, a curve in the second section and a straight again in the third section. We define the time $\tilde{t}_j$ at which the vehicle crosses the finish line at the j-th iteration as

$$\tilde{t}_j = \min\{t \in \mathbb{Z}_{0+} : x_t^j \in \mathcal{L}\}. \tag{34}$$

The track borders are described by the box constraint $|e_y| \leq 1.6m$. Note that we assumed that the road constraint on the lateral distance e_y takes into account the width of the vehicle.

The initial feasible trajectory and the locally optimal trajectory that the controller converges to are shown in Figure 3. It is seen that the vehicle cuts the corner performing a trajectory with a constant radius of curvature. In particular, we see that the controller saturates the road constraint in the second section of the path, which represents the curve (Fig. 6). This behavior was shown to be the optimal solution for the minimum time problem racing problems [14].

Figure 4 depicts the velocity profile along the trajectory. The controller slows down just before entering corner, speeds up right after the midpoint of the curve. This behavior is consistent with racing driver performance [6] [15].

The controller successfully decreases the traveling time from the starting point, x_S, to the finish line in Figure 2. Furthermore, we see in Figure 5 that the travelling time is decreasing at each iteration until the LMPC (31) and (33) reaches convergence.

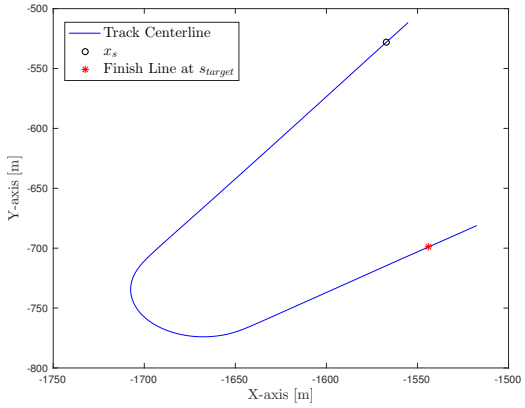

Fig. 2. Path used for testing the proposed control logic.

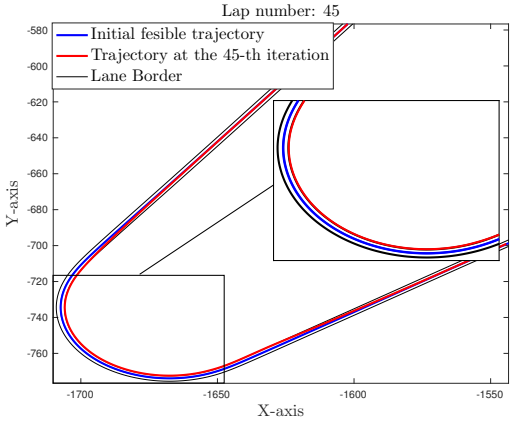

Fig. 3. Steady state trajectory of the proposed control strategy on the $X-Y$ plane.

Figure 6 shows the evolution of the vehicle longitudinal velocity and lateral distance from the center-line. The controller correctly understands the benefit of cutting the curve until it saturates the road constraints. Moreover, the controller brakes until the midpoint of the curve to prevent the vehicle to drift out of the track.

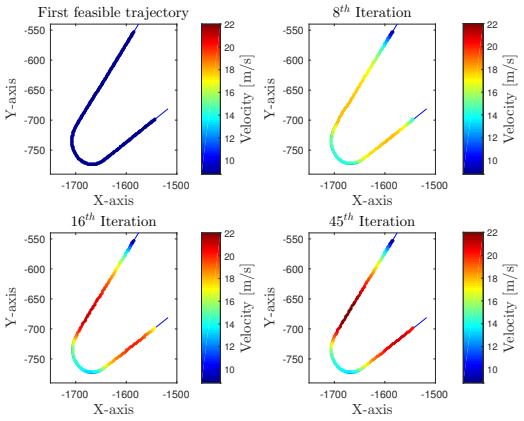

Fig. 4. Evolution of the velocity profile over the iterations.

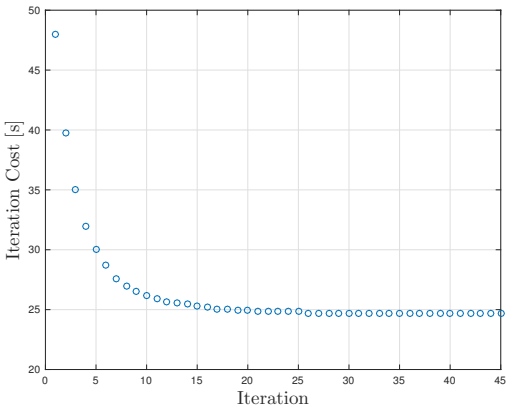

Fig. 5. Evolution of the iteration cost over the iterations. We notice that the LMPC coupled with the proposed system identification technique decreases the travelling time at each iteration.

We analyse the one step prediction errors at time t of the j-th iteration,

$$\nabla i_t^j = i_{t+1|t}^{*,j} - i_{t+1}^j, \ \forall i \in [v_x, v_y, \dot{\psi}] \tag{35}$$

which quantifies the model mismatch between the real model (1) and the learnt model (15). Figure 7 reports the one step prediction error for the 45-th iteration, compared with the actual system state. We notice that the proposed distance-based identification technique correctly identifies the system dynamics within an acceptable tolerance, in particular we

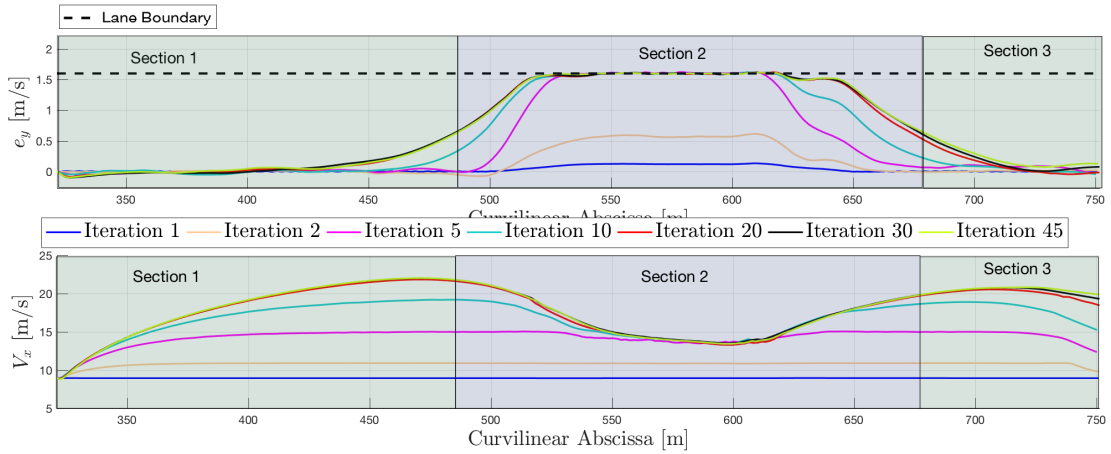

Fig. 6. Evolution of the closed-loop trajectory over the iterations

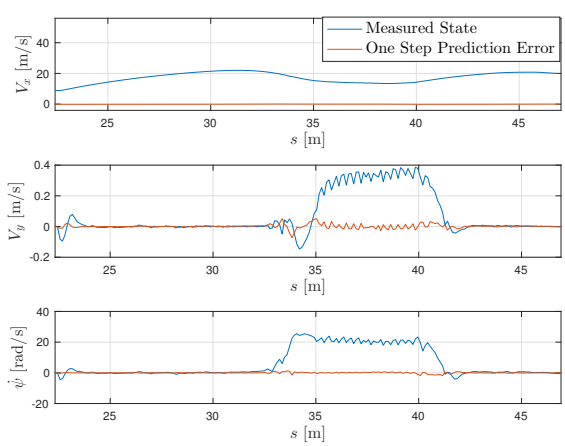

Fig. 7. Comparision between the prediciton error and the closed-loop trajectory.

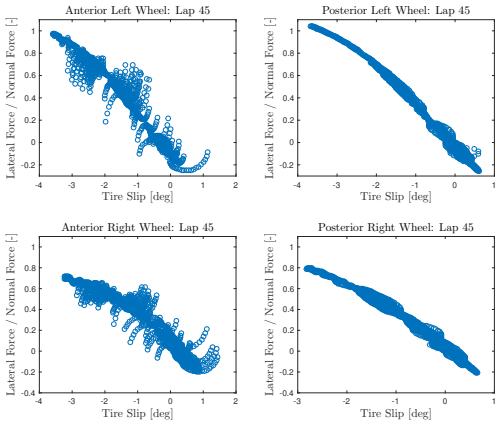

Fig. 8. Analysis of the tire forces. We notice that the left wheels are operating at their handling capability.

have,

$$\max_t |\nabla v_{x_t}^j| = 0.0587 \ m/s$$
$$\max_t |\nabla v_{y_t}^j| = 0.0511 \ m/s$$
$$\max_t |\nabla \dot{\psi}_t^j| = 0.0224 \ rad/s.$$

Finally, Figure 8 shows that LMPC (31) and (33) saturates the tire capabilities of the left front and rear tire. Therefore, we conform that the proposed control logic is able to identify the vehicle's performance limit and operate the vehicle at the limit of its handling capability.

VII. Conclusions

In this paper, a learning nonlinear 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 time varying approximation safe set and a terminal cost, learnt from previous iterations, are presented. Moreover, we coupled the LMPC with a distance-based identification method that allows the controller to operate the vehicle at the limit of its handling capability. We tested the proposed control logic in simulation with the high fidelity simulator CarSim and we showed that the controller is able to identify the vehicle dynamics and to operates the vehicle close to the limit of its

handling capability.

VIII. APPENDIX I

The model used in the identification techniques presented in Section IV is described. We assumed that the known dynamics in (15) is given by the kinematic model in the error reference frame,

$$\bar{g}_t^j(x_i^j, u_i^j) = x_i^j + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \left[\dot{\psi}_t - \frac{v_{x_t} cos(e_{\psi_t}) - v_{y_t} sin(e_{\psi_t})}{1 - e_{y_t}\kappa(s_t)}\kappa(s_t)\right] dt \\ (v_{x_k} sin(e_{\psi_t}) + v_{y_t} cos(e_{e_{\psi_t}}))dt \\ \frac{v_{x_k} cos(e_{\psi_t}) - v_{y_k} sin(e_{\psi_t})}{1 - e_{y_k}\kappa(s_k)} dt \end{bmatrix}, \tag{37}$$

where $\kappa(s)$ is the angle of the tangent vector to the path at the curvilinear abscissa s_t. The states e_{y_t} and e_{ψ_t} represent the lateral distance and heading angle error between the vehicle and the path. v_{x_t}, v_{y_t} and $\dot{\psi}_t$ are the vehicle longitudinal velocity, lateral velocity and yaw rate, respectively.
Furthermore, the linear regressor in (16) is used to identify the longitudinal and lateral dynamics and it is given by

$$g_t^j(x_i^j, u_i^j) = \begin{bmatrix} g_{t,1}^j(x_i^j, u_i^j) \\ g_{t,2}^j(x_i^j, u_i^j) \\ g_{t,3}^j(x_i^j, u_i^j) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{38}$$

where the feature vectors in (17) are defined as

$$\gamma_{1,t}^j = [v_{x_t}^j, v_{y_t}^j \dot{\psi}_t^j, a_t^j] \qquad \in \mathbb{R}^3 \tag{39a}$$

$$\gamma_{2,t}^j = \left[\frac{v_{y_t}^j}{v_{x_t}^j}, \dot{\psi}_t^j v_{x_t}^j, \frac{\dot{\psi}_t^j}{v_{x_t}^j}, \delta_t^j\right], \quad \in \mathbb{R}^4 \tag{39b}$$

$$\gamma_{3,t}^j = \left[\frac{\dot{\psi}_t^j}{v_{x_t}^j}, \frac{v_{y_t}^j}{v_{x_t}^j}, \delta_t^j\right], \qquad \in \mathbb{R}^3. \tag{39c}$$

Note that the features are chosen based on the dynamic bicycle model, for further details we refer to [10].

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] 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.

[4] 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.

[5] 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, pp. 2753–2758.

[6] 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.

[7] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks," *Submitted to TAC, available at https://arxiv.org/abs/1609.01387*, 2016.

[8] 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.

[9] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4029–4036.

[10] 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. Micaelli and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," Ph.D. dissertation, INRIA, 1993.

[12] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "User's guide for npsol (version 4.0): A fortran package for nonlinear programming." DTIC Document, Tech. Rep., 1986.

[13] U. Rosolia, S. De Bruyne, and A. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–16, 2016.

[14] E. Velenis and P. Tsiotras, "Minimum time vs maximum exit velocity path optimization during cornering," in *2005 IEEE international symposium on industrial electronics*, 2005, pp. 355–360.

[15] A. Rucco, G. Notarstefano, and J. Hauser, "Computing minimum lap-time trajectories for a single-track car with load transfer," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 6321–6326.