Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 3142–3147

Learning Model Predictive Control for Iterative Tasks: A Computationally Efficient Approach for Linear System

Ugo Rosolia * **Francesco Borrelli** *

* *University of California at Berkeley , Berkeley, CA 94701, USA (e-mail: {ugo.rosolia, fborrelli}@berkeley.edu).*

Abstract: A Learning Model Predictive Controller (LMPC) for linear system is presented. The proposed controller builds on previous work on nonlinear LMPC and decreases its computational burden for linear system. The control scheme is reference-free and is able to improve its performance by learning from previous iterations. A convex safe set and a terminal cost function are used in order to guarantee recursive feasibility and non-increasing performance at each iteration. The paper presents the control design approach, and shows how to recursively construct the convex terminal set and the terminal cost from state and input trajectories of previous iterations. Simulation results show the effectiveness of the proposed control logic.

Keywords: Learning, Model Predictive Control, LMPC, Convex Optimization

1. INTRODUCTION

Iterative Learning Control (ILC) studies control design for autonomous systems performing repetitive tasks Bristow et al. (2006); Lee and Lee (2007); Wang et al. (2009). One task execution is often referred to as "iteration" or "trial". 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 Bristow et al. (2006); Lee and Lee (2007). The tracking error from the previous iterations is used to improve the tracking performance of the closed loop system. Different strategies have been proposed to guarantee zero tracking error of the closed loop system Bristow et al. (2006); Lee and Lee (2007); Wang et al. (2009).

Several control frameworks which combine ILC and MPC strategies have been proposed in literature, Subbaraman and Benosman (2016); Lee and Lee (2000); Lee et al. (2000). In the classical ILC approach the goal of the controller is to track a reference trajectory, however, in some application such has autonomous racing Sharp and Peng (2011); Rucco et al. (2015) or for some manipulation tasks Tamar et al. (2016), it may be challenging to generate a priori a reference trajectory that maximize the system performance. For this reason, a very recent work Tamar et al. (2016) proposed a reference-free ILC scheme. The authors used a MPC controller with a terminal cost that allows to consider the long term planning. This terminal cost is computed using a neural network trained on data generated by offline simulations. The authors were able to improve the system performance over iterations. However, no guaranties about stability, recursive feasibility and performance improvement are provided.

Our objective is to design a reference-free iterative control strategy for linear system able to learn from previous iterations. 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 realized closed loop system trajectory. The iteration cost shall not increase over the iterations and state and input constraints shall be satisfied. 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 Garcia et al. (1989). However, the receding horizon nature can lead to infeasibility and it does not guaranty improved performance at each iteration Mayne et al. (2000).

The contribution of this paper is the following. We present an extension to the learning MPC for iterative control task in Rosolia and Borrelli (2016). In particular, we introduce a new formulation for linear system that drastically reduces the computation burden of the controller without compromising the guaranties of the learning MPC. We show how to design a convex safe set and a terminal cost function in order to guarantee: *(i):* [*asymptotic stability*], the closed loop system converges asymptotically to the equilibrium point, *(ii):* [*persistent feasibility*], state and input constraints are satisfied if they were satisfied at iterations $j-1$ *(iii):* [*performance improvement*], the j-th iteration cost does not increase compared with the j-1-th iteration cost, *(iv):* [*global optimality*], if the steady state system converges to a closed-loop trajectory as the number of iterations j goes to infinity, then that closed-loop trajectory is globally optimal. We emphasize that *(i)*-*(ii)* are standard MPC design requirement and *(iii)*-*(iv)* are the core contribution of this work.

This paper is organized as follows: in Section II we introduce the notation used throughout the paper. Then, we define the convex safe set and the terminal cost function used in the design of the learning MPC. Section III describes the control design. We show the recursive feasibility and stability of the control logic and, afterwards, we prove

10.1016/j.ifacol.2017.08.324

the convergence properties. Finally, in Section IV we test the proposed control logic on an infinite horizon linear quadratic regulator and we compare the computational efficiency with the learning MPC from Rosolia and Borrelli (2016).

2. PROBLEM FORMULATION

Consider the discrete time system

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the system state and input, respectively, subject to the constraints

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

where $\mathcal{X}$ and $\mathcal{U}$ are convex sets.

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, respectively. 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:

$$J^*_{0\to\infty}(x_S) = \min_{u_0,u_1,\dots} \sum_{k=0}^{\infty} h(x_k, u_k) \tag{5a}$$
$$\text{s.t. } x_{k+1} = Ax_k + Bu_k,\ \forall k \geq 0 \tag{5b}$$
$$x_0 = x_S, \tag{5c}$$
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \forall k \geq 0 \tag{5d}$$

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, jointly convex 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)

$$x_F = Ax_F. \tag{7}$$

Next we introduce the definition of the convex safe set and of the terminal cost. Both will be used later to guarantee stability and feasibility of the learning MPC for linear system.

2.1 Convex Safe Set

In the following we recall the definition of the sampled Safe Set from Rosolia and Borrelli (2016) which is necessary to construct the convex Safe Set used in the learning MPC for linear system.

The definition of the *sampled Safe Set* exploits the iterative nature of the control task to define an invariant control set, using the realized system trajectories. At the j-th iteration the sampled safe set, $\mathcal{SS}^j$, is defined as

$$\mathcal{SS}^j = \left\{ \bigcup_{i\in M^j} \bigcup_{t=0}^{\infty} x_t^i \right\}. \tag{8}$$

$\mathcal{SS}^j$ is the collection of all state trajectories at iteration i for $i \in M^j$. M^j in equation (8) is the set of indexes k associated with successful iterations k for $k \leq j$, defined as:

$$M^j = \Big\{ k \in [0, j] : \lim_{t\to\infty} x_t^k = x_F \Big\}. \tag{9}$$

Moreover, as $\mathcal{X}$ and $\mathcal{U}$ are convex, for each convex combination of the elements in $\mathcal{SS}^j$ we can find a control sequence that steers the system (1) to x_F. Therefore, the *convex Safe Set*, defined as

$$\mathcal{CS}^j = \text{Conv}(\mathcal{SS}^j) = \Big\{ \sum_{i=1}^{|\mathcal{SS}^j|} \alpha_i z_i : \alpha_i \geq 0,\ \sum_{i=1}^{|\mathcal{SS}^j|} \alpha_i = 1,\quad z_i \in \mathcal{SS}^j \Big\}, \tag{10}$$

is a control invariant set. Note that $|\mathcal{SS}^j|$ is the cardinality of $\mathcal{SS}^j$. For further details on control invariant set we refer to Borrelli (2003).

From (9) we have that $M^i \subseteq M^j, \forall i \leq j$, which implies that

$$\mathcal{CS}^i \subseteq \mathcal{CS}^j, \forall i \leq j. \tag{11}$$

2.2 Terminal 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

$$J^j_{t\to\infty}(x_t^j) = \sum_{k=0}^{\infty} h(x_{t+k}^j, u_{t+k}^j), \tag{12}$$

where $h(\cdot,\cdot)$ is the stage cost of problem (5). We define the j-th iteration cost as the cost (12) of the j-th trajectory at time $t = 0$,

$$J^j_{0\to\infty}(x_0^j) = \sum_{k=0}^{\infty} h(x_k^j, u_k^j). \tag{13}$$

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

Remark 1. In equations (12)-(13), x_k^j and u_k^j are the realized state and input at the j-th iteration, as defined in (3).

Finally we define the, barycentric function (Jones and Morari (2010))

$$P^j(x) = \begin{cases} p^{j,*}(x) & \text{If } x \in \mathcal{CS}^j \\ +\infty & \text{If } x \notin \mathcal{CS}^j \end{cases} \tag{14}$$

where