Non-tracking Type Iterative Learning Control Based on Economic MPC

Yushen Long[1], Lihua Xie[2] and Shuai Liu[3]

Abstract

Most existing iterative learning control (ILC) algorithms were designed to improve tracking performance with respect to a given trajectory over a fixed time period $[0, T]$. In this paper, we design two iterative learning based economic model predictive controllers (ILEMPC) for repetitive tasks where no target trajectory is available. The controller is able to search for suboptimal trajectories with good performance by exploiting information from previous experience. Compared with existing works, the objective function is not assumed to be positive definite so it is not limited to the tracking problem but can represent more general economic performance index. The controller can learn from the previous closed-loop trajectory, resulting in a performance which is guaranteed to be no worse than the previous one. Under some standard assumptions in model predictive control, the recursive feasibility of the algorithms is ensured. We show that the fixed operation time algorithm can guarantee performance improvement even without the dissipative assumption. By allowing the operation time to vary, the flexible operation time algorithm can balance the operation time and the system performance if dissipative assumption is satisfied. For both algorithms, each iteration is guaranteed to be completed within a uniformly bounded time duration.

I. Introduction

In this paper, we consider the following nonlinear system:

$$x_j(k+1) = f(x_j(k), u_j(k)),\ x(0) = x^0, \tag{1}$$

[1]Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632. {Long_Yushen}@i2r.a-star.edu.sg

[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. {elhxie}@ntu.edu.sg

[3]School of Control Science and Engineering, Shandong University, Jinan 250061, China. liushuai@sdu.edu.cn

where $x \in \mathbb{X} \subset \mathbb{R}^n$ is the state and $u \in \mathbb{U} \subset \mathbb{R}^m$ is the control input, $\mathbb{X}$ and $\mathbb{U}$ are compact, j is the iteration index and k is the time instant. Such a model is used to describe the dynamics of systems for repetitive tasks in robot manipulation [25], batch processes [13], stroke rehabilitation [14], and so on. For each iteration j, the system starts from time instant $k = 0$ and a fixed initial state x^0 and runs until the task is completed. Then the time instant and system state are reset as $k = 0$ and x^0, respectively and the system goes to the next iteration.

Iterative learning control (ILC) [3] has been studied in past few decades for such a system. In traditional ILC, usually there are six postulates [4]:

- P1: Every iteration ends in a fixed time of duration $T > 0$.
- P2: There is a desired trajectory $x_d(k)$ given *a priori* over $[0, T]$.
- P3: Every iteration starts from the same initial state x^0.
- P4: System dynamics is invariant throughout iterations.
- P5: $x_j(k)$ is measurable.
- P6: There exists a unique feasible $u_d(k)$ leading to the desired state trajectory $x_d(k)$.

The objective of a traditional ILC is to let system (1) perfectly track the desired trajectory $x_d(k)$ as the iteration index j goes to infinity, i.e., $\lim_{j\to\infty} x_j(k) = x_d(k)$. The main methodology of a traditional ILC is to use trajectories of previous iterations to correct the control input of current iteration, e.g., $u_j(k) = g(x_j(k), e_{j-1}(k))$, where $e_{j-1}(k) = x_{j-1}(k) - x_d(k)$, so that $e_j(k)$ goes to zero as j goes to infinity. Apparently, the ultimate performance of system (1) is determined by the performance of x_d. However, to find an optimal x_d we need to solve a constrained optimal control problem over $[0, T]$, which may be difficult when T is large.

The main objective of this paper is to propose ILC algorithms for system (1) **without** a desired trajectory given *a priori* but can explore feasible trajectories by itself and find some sub-optimal ones. Since the objective of system (1) is no longer to track a given trajectory, unlike a traditional ILC which mainly considers the tracking error as performance index, we can optimize more general economic performance index during the learning procedure.

On the other hand, model predictive control (MPC) is known for its capability to handle hard constraints on state and control and optimize system performance. A recent survey in [24] indicates that model predictive control (MPC) is the second most successful control technology in industry and the most successful advanced control technology if one excludes PID control from the advanced ones. A lot of existing literature have contributed to the theoretical analysis

[18], [22], [19] and practical application [21], [1], [26] of MPC.

The combination of ILC and MPC will be the main methodology of this paper due to their aforementioned features. The combination of ILC and MPC has been studied in a few papers. In these works, repetitive tasks defined as tracking a given reference trajectory over a fixed time duration $[0, T]$ under a fixed initial condition are considered. In [13], the authors propose an ILMPC for batch processes, which is based on a time-varying MIMO linear model. Experiments on a nonlinear batch reactor system show that it outperforms the traditional PID controller and ILC controller. In [12] the authors prove that for linear systems, the tracking error of the controller in [13] converges to zero as the number of iterations goes to infinity. In [11], the authors further extend the analysis to linear time-varying systems with disturbances. The tracking errors of previous iterations are explicitly incorporated into the control input of current iteration in order to minimize the tracking error. An observer is also designed when system is subject to deterministic or stochastic disturbances. In [20], an ILMPC is formulated based on an incremental state-space model. It is proved that for a disturbance-free linear system, the tracking error converges to zero. An extension to cases with disturbances is also discussed and tested by numerical examples. An ILMPC for nonlinear systems is proposed in [5] based on a series of time-varying linear models along the state trajectory. Assuming that the desired reference is reachable, the authors prove that tracking error converges to zero under some mild assumptions. The non-linearity of system model is handled by a T-S fuzzy model in [15]. The disturbance is rejected by an MPC and an ILC is designed to minimize the accumulative tracking error and excessive input movement. The tracking error along iterations is also proved to be convergent to zero. Different from the aforementioned papers, in [23], the authors do not assume that the reference trajectory is known. A 'database' is constructed by using trajectories of previous iterations and the terminal condition of MPC controller is formulated by choosing the best trajectory from the 'database'. Under some convexity conditions, the authors prove that if the trajectory converges as iteration index goes to infinity, then the limit trajectory is the optimal solution of a quasi-infinite horizon optimal control problem. However, this result is given in an asymptotic manner, which may not be practical for iterative learning control since for each iteration, the operation has to be terminated in finite time.

The MPC formulations in the aforementioned references are all trying to minimize the tracking error, with respect to a desired trajectory [12], [11], [5] or an equilibrium [23]. However, such a tracking type cost function does not necessarily represent the actual economic cost involved in plant operation. Economic MPC (EMPC) [6] has been studied in recent years as a tool to trade-off system behavior between two extreme cases: pure economic optimization and pure tracking problem. Compared with standard stabilizing MPC, where to ensure the stability, the stage cost must be chosen as a positive definite function with respect to the equilibrium, EMPC allows an arbitrary stage cost function, and hence performance indexes other than tracking error could be handled. As a result, EMPC could be adopted in more real world applications where tracking error is not the main consideration, such as process industry [7], water distribution systems [28] and smart buildings [17]. However, most of these works focus on a single batch operation. For repetitive tasks, EMPC is rarely considered since most ILC aim to reduce the tracking error, which is naturally positive definite and the concept of EMPC is hard to be applied.

In this paper we are going to design ILC algorithms **without** any desired trajectory. We aim to remove the design phase of the desired trajectory. The designed controller can search trajectories with better performance starting from an initial feasible trajectory by itself. The designed algorithms are able to learn from previous iterations to improve the closed-loop system performance, which is not necessarily the tracking error but can be more general economic performance index.

The contribution of this paper is summarized as follows:

1) We propose two novel ILEMPC algorithms for systems executing repetitive tasks. Without a given desired trajectory to be tracked, system (1) will be able to learn trajectories with better and better performance by itself. Compared with [23], we do not assume that the optimum of the stage cost is the steady state, which allows us to optimize economic cost of the plant directly. Furthermore, the algorithms are guaranteed to drive system states to the equilibrium in finite time so its practicality is ensured, while in [23] the results are given in an asymptotic manner, which is not consistent with the iterative learning setup.

2) We show that the recursive feasibility is guaranteed for the proposed MPC controller formulation. For the fixed operation time algorithm, even **without** the commonly used dissipative assumption, we show the performance improvement is guaranteed. For the flexible operation time algorithm, we prove that, under similar assumptions in [2], the performance improvement is also guaranteed. Furthermore, the algorithm can balance the operation time and the system performance via an additional penalty term.

The rest of this paper is organized as follows. In Section II we formulate the iterative control problem, introduce the performance index to be considered and give the design objective. In Section III, we propose two algorithms and analyze their performance respectively. In Section IV, three numerical examples are given to illustrate the effectiveness of the proposed algorithms. In Section V, some conclusions will be drawn.

Some remarks on notations are introduced as follows. We use $\mathbb{R}$ to denote the set of real numbers. $\mathbb{R}^n$ and $\mathbb{N}$ denote n-dimensional Euclidean space and the set of natural numbers, respectively. For a vector $x \in \mathbb{R}^n$, $\|x\|_2$ denotes its 2-norm.

II. Problem Formulation and Preliminary

Suppose that at the very beginning we have a feasible state and control sequence:

$$x_0(0), \dots, x_0(T);$$

$$u_0(0), \dots, u_0(T),$$

where $x_0(0) = x^0$, $x_0(i) \in \mathbb{X}$, $u_0(i) \in \mathbb{U}, \forall i = 0, \dots, T$, $x_0(T) = x_s$, $u_0(T) = u_s$ and $x_s = f(x_s, u_s)$. The subscript 0 denotes the initial trajectory and in the following part of this paper, subscript j will be used to denote the j-th iteration.

The stage cost $l(x, u)$ satisfies the following assumption:

Assumption 2.1: $l(x, u)$ is continuous in $\mathbb{X} \times \mathbb{U}$.

We assume that system (1) executes the same operation over and over again. The task of each iteration is to reach the steady state x_s. After the steady state x_s is reached, the system state is reset as the fixed initial condition x^0 to start the next iteration. Thus, in what follows, we assume that for each iteration the initial condition is x^0.

Remark 2.1: In this manuscript, for simplicity, we assume a fixed initial condition and do not consider uncertainties such as external disturbances explicitly. However, the proposed algorithm can be easily extended to cases with bounded uncertainty on the initial condition and/or bounded external disturbance by using the tube-based approach in [10] and [29]. In a disturbed case, the convergence and performance improvement hold for the nominal trajectory while the exact trajectory evolves around the nominal one.

The objective of a traditional iterative learning control is to track a given trajectory over a finite time horizon $[0, T]$. In this paper, we do not let the system to track a given trajectory but to learn trajectories from the initial state to the equilibrium with better and better performance. In what follows we will propose two different control strategies. The first one can drive the state to the equilibrium no later than $k = T$ while improving system performance. The second one has more flexibility on the operation time of each iteration: the operation time is uniformly bounded for each iteration but can be larger than T. In the second strategy, the operation time and the system performance will be balanced via an additional penalty term in the optimization problem.

III. Algorithm Design and Analysis

In this section, two different strategies will be designed and analyzed. The first algorithm guarantees that each iteration ends no later than time instant T. It has less flexibility on performance improvement but very few assumptions on the system are required. The second algorithm guarantees that each iteration ends within a finite time which can be longer than T. It has more flexibility on performance improvement but requires the system to be dissipative.

A. Fixed operation time case

We first consider the fixed operation time case where the state should be driven to the equilibrium no later than time instant T.

We propose the following iterative learning economic MPC to improve the performance: For the j-th iteration, at time instant $k = 0, \ldots, T-1$, the following optimization problem is solved:

Problem 1

$$\min_{u_j(k|k),\ldots,u_j(k+N_k-1|k)} \sum_{i=k}^{k+N_k-1} l(x_j(i|k), u_j(i|k))$$

subject to

$$\begin{aligned} x_j(i+1|k) &= f(x_j(i|k), u_j(i|k)), \\ x_j(i|k) &\in \mathbb{X}, \\ u_j(i|k) &\in \mathbb{U}, \\ x_j(k+N_k|k) &= x_{j-1}(k+N_k), \\ x_j(k|k) &= x_j(k),\ i = k, \ldots, k+N_k-1, \end{aligned} \tag{2}$$

where the prediction horizon $N_k = \begin{cases} N, \text{ if } N+k \leq T \\ T-k, \text{ otherwise,} \end{cases}$ and N is a positive integer strictly less than T.

Denote the optimal solution and the corresponding state trajectory as

$$u_j^*(k|k),\ u_j^*(k+1|k),\ldots,\ u_j^*(k+N_k-1|k),$$

$$x_j^*(k|k),\ x_j^*(k+1|k),\ldots,\ x_j^*(k+N_k-1|k),\ x_j^*(k+N_k|k),$$

and optimal value function as $V_j^*(x_j(k)) = \sum_{i=k}^{k+N_k-1} l(x_j^*(i|k), u_j^*(i|k))$. The control input to be fed to the plant is $u_j(k) = u_j^*(k|k)$.

Constraint (2) is the key to the proposed algorithm. It incorporates information from the previous iteration into the calculation of current control input. Under the constraint, we can construct a feasible solution, which leverages the state trajectories from the previous iterations, for the optimization problem. From later analysis, one can observe that it helps build a connection between two consecutive iterations and guarantee performance improvement.

Theorem 3.1: Suppose that an initial feasible state and control sequence is available. Then Problem 1 is feasible for any iteration j and any time instant k. Furthermore, $x_j(T) = x_s,\ \forall j \in \mathbb{N}$.

Proof: Suppose that after the $(j-1)$-th iteration, a feasible state and control sequence $x_{j-1}(k)$ and $u_{j-1}(k),\ k = 0,\ldots,T$ is obtained. Then for the j-th iteration, at time instant 0, the following state and control sequence is feasible, by the feasibility of $x_{j-1}(k)$ and $u_{j-1}(k),\ k = 0,\ldots,T$:

$$\begin{aligned} x_j(k|0) &= x_{j-1}(k),\ k = 0,\ldots,N, \\ u_j(k|0) &= u_{j-1}(k),\ k = 0,\ldots,N-1. \end{aligned}$$

Suppose that at time instant k of the j-th iteration, Problem 1 is feasible. We discuss the following two cases separately.

Case 1: $N_k + k + 1 \leq T$.

Then in this case, $N_k = N_{k+1} = N$.

By the terminal constraint (2), we have $x_j^*(k+N|k) = x_{j-1}(k+N)$. Therefore, for time instant $k+1$, we can construct the following candidate solution:

$$u_j^*(k+1|k),\ldots,\ u_j^*(k+N-1|k),\ u_{j-1}(k+N)$$

and the corresponding state trajectory

$$x_j^*(k+1|k), \ldots,\ x_j^*(k+N-1|k),\ x_{j-1}(k+N),\ x_{j-1}(k+N+1),$$

which are feasible by the feasibility of the $(j-1)$-th iteration.

Case 2: $N_k + k = T$.

In this case, $N_{k+1} = N_k - 1$ and $x_{j-1}(k+N_k) = x_{j-1}(T) = x_s$.

By the terminal constraint (2), we have $x_j^*(k+N_k|k) = x_{j-1}(T) = x_s$. Therefore, for time instant $k+1$, we can construct the following candidate solution:

$$u_j^*(k+1|k), \ldots,\ u_j^*(T-1|k)$$

and the corresponding state trajectory

$$x_j^*(k+1|k), \ldots,\ x_j^*(T|k) = x_s,$$

which are feasible by the feasibility of the $(j-1)$-th iteration. Note that when $k = T-1$, $x_j(T) = x_j(T|k) = f(x_j(k), u_j(k)) = x_{j-1}(T) = x_s$.

The proof is completed by induction. ■

Denote $J_j = \sum_{i=0}^{T-1} l(x_j(i), u_j(i))$.

Theorem 3.2: If Assumption 2.1 holds, then $J_{j+1} \leq J_j,\ \forall j \in \mathbb{N}$.

Proof: By the continuity of $l(x,u)$, $L(x,u)$ is upper bounded in $\mathbb{X} \times \mathbb{U}$.

Similar to Theorem 3.1, we discuss the following two cases separately.

Case 1: $N_k + k + 1 \leq T$.

Suppose that at the j-th iteration and time instant k, we have the optimal solution of Problem 2:

$$u_j^*(k|k), \ldots, u_j^*(k+N-1|k)$$

and the corresponding state trajectory:

$$x_j^*(k|k), \ldots, x_j^*(k+N|k).$$

Consider the following feasible solution for time instant $k+1$ as in Theorem 3.1:

$$u_j^*(k+1|k), \ldots, u_j^*(k+N-1|k), u_{j-1}(k+N)$$

and the corresponding state trajectory:

$$x_j^*(k+1|k), \ldots, x_j^*(k+N|k), x_{j-1}(k+N+1).$$

So, the optimal value function of $x_j(k+1)$ satisfies that

$$\begin{aligned} V_j^*(x_j(k+1)) &\leq \sum_{i=k+1}^{k+N-1} l(x_j^*(i|k), u_j^*(i|k)) + l(x_{j-1}(k+N), u_{j-1}(k+N)) \\ &= V_j^*(x_j(k)) + l(x_{j-1}(k+N), u_{j-1}(k+N)) - l(x_j(k), u_j(k)). \end{aligned}$$

Taking summation of the above inequalities from $k=0$ to $k=T-N-1$ leads to that

$$V_j^*(x_j(T-N)) - V_j^*(x_j(0)) \leq \sum_{i=N}^{T-1} l(x_{j-1}(i), u_{j-1}(i)) - \sum_{i=0}^{T-N-1} l(x_j(i), u_j(i)). \quad (3)$$

Case 2: $N_k + k = T$.

In this case, we have the optimal solution of Problem 2 at time instant k as:

$$u_j^*(k|k), \ldots, u_j^*(T-1|k)$$

and the corresponding state trajectory:

$$x_j^*(k|k), \ldots, x_j^*(T|k) = x_s.$$

And the feasible solution for time instant $k+1$ can be constructed as:

$$u_j^*(k+1|k), \ldots, u_j^*(T-1|k)$$

and the corresponding state trajectory:

$$x_j^*(k+1|k), \ldots, x_j^*(T|k) = x_s.$$

So, the optimal value function of $x_j(k+1)$ satisfies that

$$\begin{aligned} V_j^*(x_j(k+1)) &\leq \sum_{i=k+1}^{T} l(x_j^*(i|k), u_j^*(i|k)) \\ &= V_j^*(x_j(k)) - l(x_j(k), u_j(k)) \end{aligned}$$

Taking summation of the above inequalities from $k=T-N$ to $k=T-2$ leads to that

$$V_j^*(x_j(T-1)) - V_j^*(x_j(T-N)) \leq - \sum_{i=T-N}^{T-2} l(x_j(i), u_j(i)). \quad (4)$$

By combining (3) and (4) together we have

$$V_j^*(x_j(T-1)) - V_j^*(x_j(0)) \leq \sum_{i=N}^{T-1} l(x_{j-1}(i), u_{j-1}(i)) - \sum_{i=0}^{T-2} l(x_j(i), u_j(i)). \quad (5)$$

Note that $V_j^*(x_j(T-1)) = l(x_j(T-1), u_j(T-1))$. Rearranging (5) results in

$$\begin{aligned} J_j &\leq V_j^*(x_j(0)) + \sum_{i=N}^{T-1} l(x_{j-1}(i), u_{j-1}(i)). \\ &\leq \sum_{i=0}^{T-1} l(x_{j-1}(i), u_{j-1}(i)) \\ &= J_{j-1}, \end{aligned}$$

where in the second inequality we use $V_j^*(x_j(0)) \leq \sum_{i=0}^{N-1} l(x_{j-1}(i), u_{j-1}(i))$. ■

Remark 3.1: In this fixed operation time algorithm, the performance improvement and the finite-time convergence to the equilibrium are completely guaranteed by the terminal constraint (2) and the shrinking horizon strategy. Therefore, only the boundedness of the objective function is required. Neither positive definiteness nor dissipativeness is necessary for the desired results.

B. Flexible operation time case

In the previous approach, the system state takes at most T time steps to reach the equilibrium. In this subsection, we will propose another approach which allows each iteration to take more than T time step to reach the equilibrium so that the performance and the operation time can be balanced. To this end, we introduce the following stage cost:

$$h(x, u) = l(x, u) + \alpha \mathbb{I}(x, u), \tag{6}$$

where $\mathbb{I}(x,u) = \begin{cases} 1, & \text{when } (x,u) \neq (x_s, u_s), \\ 0, & \text{otherwise,} \end{cases}$ and $\alpha > 0$ is a positive constant.

Denote N as the prediction horizon and $T_j = \inf_{k \geq T_{j-1}-N} \{(x_j(k), u_j(k)) = (x_s, u_s)\}$. Given a feasible trajectory of the $(j-1)$-th iteration, we define virtually infinite sequences of state and control as follows:

$$\tilde{x}_{j-1}(k) = \begin{cases} x_{j-1}(k), & \text{when } k \leq T_{j-1}, \\ x_s, & \text{when } k > T_{j-1}, \end{cases}$$

and

$$\tilde{u}_{j-1}(k) = \begin{cases} u_{j-1}(k), & \text{when } k \leq T_{j-1}, \\ u_s, & \text{when } k > T_{j-1}. \end{cases}$$

The control algorithm is given by solving the following optimization problem at time instant k of the j-th iteration until (x,u) reaches (x_s,u_s) and $k+N\geq T_{j-1}$:

Problem 2

$$\min_{u_j(k|k),\ldots,u_j(k+N-1|k)} \sum_{i=k}^{k+N-1} h(x_j(i|k),u_j(i|k))$$

subject to

$$\begin{aligned} x_j(i+1|k) &= f(x_j(i|k),u_j(i|k)),\\ x_j(i|k) &\in \mathbb{X},\\ u_j(i|k) &\in \mathbb{U},\\ x_j(k+N|k) &= \tilde{x}_{j-1}(k+N),\\ x_j(k|k) &= x_j(k),\ i=k,\ldots,k+N-1. \end{aligned} \tag{7}$$

Denote the optimal solution and the corresponding state trajectory as

$$u_j^*(k|k),\ u_j^*(k+1|k),\ldots,\ u_j^*(k+N-1|k),$$

$$x_j^*(k|k),\ x_j^*(k+1|k),\ldots,\ x_j^*(k+N-1|k),\ x_j^*(k+N|k),$$

and optimal value function as $V_j^*(x_j(k))=\sum_{i=k}^{k+N-1} h(x_j^*(i|k),u_j^*(i|k))$. The control input to be fed to the plant is $u_j(k)=u_j^*(k|k)$.

Theorem 3.3: Suppose that an initial feasible state and control sequence is available. Then Problem 2 is feasible for any iteration j and any time instant k.

Proof: Suppose that after the $(j-1)$-th iteration, a feasible state and control sequence $\tilde{x}_{j-1}(k)$ and $\tilde{u}_{j-1}(k),\ k\in\mathbb{N}$ is obtained. Then for the j-th iteration, at time instant 0, the following state and control sequence is feasible, by the feasibility of $\tilde{x}_{j-1}(k)$ and $\tilde{u}_{j-1}(k),\ k\in\mathbb{N}$:

$$\begin{aligned} x_j(k|0) &= \tilde{x}_{j-1}(k),\ k=0,\ldots,N,\\ u_j(k|0) &= \tilde{u}_{j-1}(k),\ k=0,\ldots,N-1. \end{aligned}$$

Suppose that at time instant k of the j-th iteration, Problem 3 is feasible. By the terminal constraint (7), we have $x_j^*(k+N|k)=\tilde{x}_{j-1}(k+N)$. Therefore, for time instant $k+1$, we can construct the following candidate solution:

$$u_j^*(k+1|k),\ldots,\ u_j^*(k+N-1|k),\ \tilde{u}_{j-1}(k+N)$$

and the corresponding state trajectory

$$x_j^*(k+1|k),\ldots,\ x_j^*(k+N-1|k),\ x_{j-1}(k+N),\ \tilde{x}_{j-1}(k+N+1),$$

which are feasible by the feasibility of the $(j-1)$-th iteration. The proof can be concluded by induction. ■

We make use of the following definition, which is standard in the economic MPC literature; see, e.g. [2].

Definition 3.1: System (1) is dissipative with respect to a supply rate $s:\ \mathbb{X}\times\mathbb{U}\rightarrow\mathbb{R}$ if there exists a continuous function $\lambda:\mathbb{X}\rightarrow\mathbb{R}$ such that:

$$\lambda(f(x,u))-\lambda(x)\leq s(x,u)$$

for all $x\in\mathbb{X}$, $u\in\mathbb{U}$.

Define two rotated stage cost $L(x,u)=l(x,u)-\lambda(f(x,u))+\lambda(x)$ and $H(x,u)=h(x,u)-\lambda(f(x,u))+\lambda(x)$, respectively.

Assumption 3.1: System (1) is dissipative with respect to the supply rate:

$$s(x,u)=l(x,u)-l(x_s,u_s).$$

Lemma 3.1: If Assumption 3.1 is satisfied, then system (1) is also dissipative with respect to the supply rate:

$$\tilde{s}(x,u)=h(x,u)-h(x_s,u_s).$$

(x_s,u_s) minimizes $H(x,u)$ and $L(x,u)$. $H(x,u)$ has a non-continuous range $\{l(x_s,u_s)\}\cup(l(x_s,u_s)+\alpha,\bar{H}]$ for some constant $\bar{H}$ and $L(x,u)$ has a continuous range $[l(x_s,u_s),\bar{L}]$ for some constant $\bar{L}$.

Proof: Since Assumption 2.1 and 3.1 are satisfied, we have $\lambda(f(x,u))-\lambda(x)\leq l(x,u)-l(x_s,u_s)$. Note that $\mathbb{I}(x,u)\geq 0$. So,

$$\begin{aligned}\lambda(f(x,u))-\lambda(x)&\leq l(x,u)+\alpha\mathbb{I}(x,u)-l(x_s,u_s)-\alpha\mathbb{I}(x_s,u_s)\\&=h(x,u)-h(x_s,u_s)\\&=\tilde{s}(x,u).\end{aligned}$$

To prove the second claim, first note that

$$\min_{(x,u)\in(\mathbb{X},\mathbb{U})}H(x,u)\leq H(x_s,u_s)=h(x_s,u_s). \tag{8}$$

On the other hand, from the dissipative property, we have

$$h(x_s, u_s) \leq h(x, u) + \lambda(x) - \lambda(f(x, u)) = H(x, u), \ \forall (x, u) \in (\mathbb{X}, \mathbb{U}),$$

which implies that

$$h(x_s, u_s) \leq \min_{(x,u)\in(\mathbb{X},\mathbb{U})} H(x, u). \tag{9}$$

Combining (8) and (9) leads to that (x_s, u_s) minimizes $H(x, u)$.

Similarly, one can also show that (x_s, u_s) minimizes $L(x, u)$. So, by letting (x, u) arbitrarily close to (x_s, u_s) and the continuity of $l(x, u)$, we have left side of the half open interval for $H(x, u)$. Also by the continuity of $L(x, u)$ and the compactness of $(\mathbb{X}, \mathbb{U})$, it is upper bounded by some constant $\bar{L}$. So we can choose $\bar{H} = \bar{L} + \alpha$ to have the right side of the half open interval for $H(X, U)$. The range for $L(x, u)$ can be demonstrated easily. ■

Since $h(x_s, u_s)$ is the minimal value of $H(x, u)$ and $h(x_s, u_s)$ is bounded, if $h(x_s, u_s) \neq 0$, we can redefine $h(x, u)$ by subtracting $h(x_s, u_s)$ from the original one. As a result, we can assume that $h(x_s, u_s) = l(x_s, u_s) = 0$, $H(x, u) \in \{0\} \cup (\alpha, \bar{H}]$ and $L(x, u) \geq 0$ in the sequel without loss of generality.

Assumption 3.2: (x_s, u_s) is the unique minimizer of $H(x, u)$.

We introduce the following auxiliary optimization problem:

Problem 3

$$\min_{u_j(k|k),\ldots,u_j(k+N-1|k)} \sum_{i=k}^{k+N-1} H(x_j(i|k), u_j(i|k))$$

subject to

$$\begin{aligned} x_j(i+1|k) &= f(x_j(i|k), u_j(i|k)), \\ x_j(i|k) &\in \mathbb{X}, \\ u_j(i|k) &\in \mathbb{U}, \\ x_j(k+N|k) &= \tilde{x}_{j-1}(k+N), \\ x_j(k|k) &= x_j(k), \ i = k, \ldots, k+N-1, \end{aligned} \tag{10}$$

where N is the prediction horizon.

Denote the optimal value function as $\tilde{V}_j^*(x_j(k)) = \sum_{i=k}^{k+N-1} H(x_j^*(i|k), u_j^*(i|k))$, where $u_j^*(i|k)$ and $x_j^*(i|k), \ j = k, \ldots, k+N-1$ are respectively the optimal control and state sequences associated with $\tilde{V}_j^*(x_j(k))$.

Lemma 3.2: For a given $x(k)$, Problem 2 and 3 have the same feasibility, i.e. if Problem 2 is feasible, then Problem 3 is also feasible and vice versa. If the problems are feasible, they have the same optimal solution(s).

The proof of this lemma follows the same line as [2] so it is omitted here. Note that in [2], the terminal constraint is the equilibrium while in our case, it is a known time-varying trajectory.

Denote $J_j = \sum_{i=0}^{T_j} h(x_j(i), u_j(i))$. Now we will proceed to prove the main results:

1) Performance improvement: $J_{j+1} \leq J_j, \ \forall j \in \mathbb{N}$;

2) Finite operation time: T_j is uniformly bounded for all $j \in \mathbb{N}$.

The main challenge here is that T_j could be different from one iteration to the next. Therefore, the direct comparison between J_j and J_{j+1} is difficult. Recall the virtually infinite sequence $\tilde{x}_j$ and $\tilde{u}_j$. There are two important properties worthy to be noted:

1) $\sum_{i=0}^{\infty} h(\tilde{x}_j(i), \tilde{u}_j(i)) = \sum_{i=0}^{T_j} h(x_j(i), u_j(i))$ so in the sequel we also use J_j to denote $\sum_{i=0}^{\infty} h(\tilde{x}_j(i), \tilde{u}_j(i))$;

2) $\tilde{x}_j$ and $\tilde{u}_j$ can also be defined by using the controller induced by Problem 4 from $k = 0$ to $k = \infty$ virtually, where Problem 4 is defined as follows:

Problem 4

$$\min_{\tilde{u}_j(k|k),\ldots,\tilde{u}_j(k+N-1|k)} \sum_{i=k}^{k+N-1} H(\tilde{x}_j(i|k), \tilde{u}_j(i|k))$$

subject to

$$\begin{aligned}
\tilde{x}_j(i+1|k) &= f(\tilde{x}_j(i|k), \tilde{u}_j(i|k)), \\
\tilde{x}_j(i|k) &\in \mathbb{X}, \\
\tilde{u}_j(i|k) &\in \mathbb{U}, \\
\tilde{x}_j(k+N|k) &= \tilde{x}_{j-1}(k+N), \\
\tilde{x}_j(k|k) &= \tilde{x}_j(k), \ i = k, \ldots, k+N-1,
\end{aligned} \tag{11}$$

where N is the prediction horizon. With some abuse of notation, we also use $\tilde{V}_j^*(\tilde{x}_j(k))$ to denote the optimal value function and $\tilde{u}_j^*(i|k)$ and $\tilde{x}_j^*(i|k), \ j = k, \ldots, k+N-1$ are respectively the optimal control and state sequences associated with $\tilde{V}_j^*(\tilde{x}_j(k))$.

In the sequel, we will show that the performance of $\tilde{x}_j$ and $\tilde{u}_j$ will be improved along the learning process. Since the performance of the virtually infinite sequence is equivalent to that of the actual finite sequence, the actual performance improvement will also be guaranteed.

Furthermore, we will show that $(\tilde{x}_j, \tilde{u}_j)$ reaches (x_s, u_s) in finite time, which in turn proves that (x_j, u_j) reaches the equilibrium in finite time so that each iteration can be terminated in finite time.

Lemma 3.3: If Assumptions 2.1, 3.1 and 3.2 are satisfied, $(\tilde{x}_j(k), \tilde{u}_j(k))$ reaches (x_s, u_s) in finite time T_j, then at least a subsequence of $(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k))$ reaches (x_s, u_s) in finite time.

Proof: By the definition of $H(x, u)$, $H(x, u)$ is upper bounded in $\mathbb{X} \times \mathbb{U}$.

Similar to the proof of Theorem 3.2, one can write that

$$\begin{aligned} & \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k+1)) - \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k)) \\ \leq \ & H(\tilde{x}_j(k+N), \tilde{u}_j(k+N)) - H(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k)). \end{aligned}$$

Taking average on both sides leads to that

$$\begin{aligned} & \frac{1}{T}\sum_{k=0}^{T-1} H(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k)) \\ \leq \ & \frac{1}{T}\sum_{k=0}^{T-1} H(\tilde{x}_j(k+N), \tilde{u}_j(k+N)) \\ & + \frac{\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(0)) - \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(T))}{T}. \end{aligned}$$

By recursive feasibility of Problem 2 and the boundedness of $H(x, u)$, we know that both $V^*_{j+1}(\tilde{x}_{j+1}(0))$ and $\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(T))$ are bounded.

Since $\tilde{x}_j(k) = x_s$ and $\tilde{u}_j(k) = u_s, \forall k \geq T_j$, $H(\tilde{x}_j(k), \tilde{u}_j(k)) = 0$ for all $k \geq T_j$. Therefore, we have $\lim_{T\to\infty} \frac{1}{T}\sum_{k=0}^{T-1} H(\tilde{x}_j(k+N), \tilde{u}_j(k+N)) = 0$ and $\lim_{T\to\infty} \frac{1}{T}\sum_{k=0}^{T-1} H(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k)) = 0$. Note that $H(\tilde{x}, \tilde{u}) = 0$ only when $(\tilde{x}, \tilde{u}) = (x_s, u_s)$ and strictly larger than some positive constant α otherwise. So, $\lim_{T\to\infty} \frac{1}{T}\sum_{k=0}^{T-1} H(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k)) = 0$ implies that at least a subsequence of $(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k))$ reaches (x_s, u_s) in finite time. ■

Then we prove that the closed-loop performance of each iteration is no worse than that of the previous one. Apparently, $\lim_{T\to\infty}\sum_{k=0}^{T} H(\tilde{x}_0(k), \tilde{u}_0(k)) < \infty$ since $(\tilde{x}_0(k), \tilde{u}_0(k))$ reaches (x_s, u_s) at $k = T$. Note that $\lim_{T\to\infty}\sum_{k=0}^{T} H(\tilde{x}_0(k), \tilde{u}_0(k)) = \lim_{T\to\infty}\sum_{k=0}^{T} h(\tilde{x}_0(k), \tilde{u}_0(k)) + \lambda(x(0)) - \lambda(x_s)$. Thus, $\lim_{T\to\infty}\sum_{k=0}^{T} h(\tilde{x}_0(k), \tilde{u}_0(k)) < \infty$. Denote $\tilde{J}_j = \lim_{T\to\infty}\sum_{k=0}^{T} H(\tilde{x}_j(k), \tilde{u}_j(k))$.

Lemma 3.4: If Assumptions 2.1 and 3.1 are satisfied, $(\tilde{x}_j(k), \tilde{u}_j(k))$ reaches (x_s, u_s) in finite time and there exists a subsequence of $\tilde{x}_{j+1}(k)$ and $\tilde{u}_{j+1}(k)$, which is denoted as $(\tilde{x}_{j+1}(k_n), \tilde{u}_{j+1}(k_n))$, reaches (x_s, u_s) in finite time and $J_j < \infty$, then $J_{j+1} \leq J_j$.

Proof: Similar to the proof of Lemma 3.3, we have

$$\begin{aligned}&\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k+1)) - \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k))\\ \leq\ & H(\tilde{x}_j(k+N), \tilde{u}_j(k+N)) - H(\tilde{x}_{j+1}(k), \tilde{u}_{j+1}(k)).\end{aligned}$$

Taking summation on both sides from $i = 0$ to $k_n - 1$ leads to that

$$\begin{aligned}&\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k_n)) - \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(0))\\ \leq\ & \sum_{i=0}^{k_n-1} H(\tilde{x}_j(i+N), \tilde{u}_j(i+N)) - \sum_{i=0}^{k_n-1} H(\tilde{x}_{j+1}(i), \tilde{u}_{j+1}(i)).\end{aligned}$$

And after rearranging it follows that:

$$\begin{aligned}&\sum_{i=0}^{k_n-1} H(\tilde{x}_{j+1}(i), \tilde{u}_{j+1}(i))\\ \leq\ & \sum_{i=0}^{k_n-1} H(\tilde{x}_j(i+N), \tilde{u}_j(i+N))\\ &+\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(0)) - \tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k_n)).\end{aligned} \tag{12}$$

Since both $\tilde{x}_{j+1}(k_n)$ and $\tilde{x}_j(k)$ reach x_s in finite time, there exists a large enough positive integer $\bar{N}$ such that for all $n \geq \bar{N}$, $\tilde{x}_{j+1}(k_n) = x_s$, $\tilde{x}_j(n) = x_s$ and $\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(k_n)) = 0$. By letting $n \to \infty$, (12) becomes

$$\tilde{J}_{j+1} \leq \tilde{J}_j - \sum_{k=0}^{N-1} H(x_j(k), u_j(k)) + \tilde{V}^*_{j+1}(x_{j+1}(0)). \tag{13}$$

Note that the right hand side of (13) is finite so this inequality is well defined. Consider Problem 3 at $k = 0$ and iteration $j+1$. By recursive feasibility of the proposed algorithm, one can observe that $\tilde{x}_j(k),\ k = 0, \ldots, N$ and $\tilde{u}_j(k),\ k = 0, \ldots, N-1$ are feasible state and control sequences for Problem 3 at $k = 0$ and iteration $j+1$. Therefore, the cost $\sum_{k=0}^{N-1} H(\tilde{x}_j(k), \tilde{u}_j(k))$ will not be smaller than the optimal one, which is $\tilde{V}^*_{j+1}(\tilde{x}_{j+1}(0))$. Combining this fact with (13), one obtains that $\tilde{J}_{j+1} \leq \tilde{J}_j,\ j \in \mathbb{N}$. The proof is completed by noticing that $\tilde{J}_j = J_j + \lambda(x(0)) - \lambda(x_s)$. ■

The first main result of this section is summarized in the following theorem.

Theorem 3.4: If Assumption 2.1, 3.1 and 3.2 are satisfied, $(\tilde{x}_0(k), \tilde{u}_0(k))$ reaches (x_s, u_s) in finite time, then

$$J_{j+1} \leq J_j$$

and $(x_j(k), u_j(k))$ reaches (x_s, u_s) in finite time, $\forall j \in \mathbb{N}$.

Proof: Applying Lemma 3.3 to $j = 0$, we have that there exists a subsequence of $x_1(k)$ and $u_1(k)$ such that $(x_1(k_n), u_1(k_n))$ reaches (x_s, u_s) in finite time.

Then we apply Lemma 3.4 to obtain that $\tilde{J}_1 \leq \tilde{J}_0 < \infty$ and $J_1 \leq J_0 < \infty$, which also imply that only finitely many terms in $(\tilde{x}_1(k), \tilde{u}_1(k))$ are not equal to (x_s, u_s). So, $(\tilde{x}_1(k), \tilde{u}_1(k))$ and $(x_1(k), u_1(k))$ reach (x_s, u_s) in finite time. The theorem can be concluded by induction. ■

Now we know that T_j is finite for all $j \in \mathbb{N}$. The second main result is given as follows.

Theorem 3.5: If Assumption 2.1, 3.1 and 3.2 are satisfied and $(x_0(k), u_0(k))$ reaches (x_s, u_s) at time $k = T_0$, then

$$T_j \leq T_0 + \frac{1}{\alpha} \sum_{i=0}^{T_0-1} L(x_0(i), u_0(i)),\ \forall j \in \mathbb{N}.$$

Proof: It has been shown that $\tilde{J}_{j+1} \leq \tilde{J}_j$ so we have $\tilde{J}_j \leq \tilde{J}_0,\ \forall j \in \mathbb{N}$. Note that $\tilde{J}_j$ can be rewritten as

$$\begin{aligned} \tilde{J}_j &= \sum_{i=0}^{T_j-1} (L(x_j(i), u_j(i)) + \alpha \mathbb{I}(x_j(i), u_j(i))) \\ &= \sum_{i=0}^{T_j-1} L(x_j(i), u_j(i)) + \alpha T_j \end{aligned}$$

and similarly

$$\tilde{J}_0 = \sum_{i=0}^{T_0-1} L(x_0(i), u_0(i)) + \alpha T_0.$$

Therefore, one has that

$$\begin{aligned} T_j &\leq T_0 + \frac{1}{\alpha}(\sum_{i=0}^{T_0-1} L(x_0(i), u_0(i)) - \sum_{i=0}^{T_j-1} L(x_j(i), u_j(i))) \\ &\leq T_0 + \frac{1}{\alpha} \sum_{i=0}^{T_0-1} L(x_0(i), u_0(i)), \end{aligned}$$

where in the second inequality we use the fact that $L(x, u) \geq 0$. ■

IV. NUMERICAL EXAMPLES

All the following examples are implemented with ICLOCS [9] and solved by IPOPT [27]. More examples can be found in [16].

We consider an inverted pendulum in [8], whose dynamics is given by

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= \sin(x_1(t)) - u(t)\cos(x_1(t)), \end{aligned}$$

where x_1 and x_2 represent the pendulum angular position and angular speed respectively. The input constraint set $\mathbb{U}$ is $\mathbb{U} = \{u \in \mathbb{R} : |u| \leq 0.5\}$ and the state constraints are $\frac{\pi}{3} \leq x_1 \leq \frac{5\pi}{3}$ and $-2 \leq x_2 \leq 2$. The initial condition is $x(0) = \begin{pmatrix} \pi \\ 0 \end{pmatrix}$. The model is discretized by forward difference approximation with sampling time interval $\delta = 0.05$s. We choose the equilibrium $x_s = (\pi - \arctan(0.5), 0)^T$ and $u_s = -0.5$ and the stage cost as $l(x,u) = \delta[(\sin(\frac{x_1}{2}), x_2)Q \begin{pmatrix} \sin(\frac{x_1}{2}) \\ x_2 \end{pmatrix} + Ru^2 + P(\|x - x_s\|_2^2 + |u - u_s|^2)]$, where $Q = \begin{pmatrix} 225 & 0 \\ 0 & 1 \end{pmatrix}$, $R = 1$ and $P = 55$ so that the dissipativity assumption is satisfied. The initial feasible trajectory is generated by solving an open-loop control problem so that it reaches the equilibrium at $k = 36$.

A. *Fixed operation time case*

We first apply the fixed operation time approach to this model. The initial feasible trajectory is extended to $T = 39$ by adding 3 more steady states at the end of the initial feasible trajectory.

The prediction horizon is chosen as $N = 20$, while the one used in [8] is $N = 60$. We ran the algorithm for 15 iterations. In Fig. 1 we show that the transient performance, which is defined as $\sum_{i=0}^{36}[l(x(i), u(i)) - l(x_s, u_s)]\delta$, improves along the learning process. In Fig. 2, the state trajectories of iterations are shown. It can be observed that the performance was improved twice along the iteration then converged to a fixed value 50.1702. In fact, the ultimate performance mainly depends on the operation time. In Fig. 3, we show the performance after 20 iterations with different operation time. When the operation time is extended to $T \geq 51$, the proposed algorithm achieves its limit performance 44.4037.

B. *Flexible operation time case*

In this example, we will apply the flexible operation time approach so that the operation time can be adjusted automatically to have better performance. We start from the initial feasible trajectory with $T = 36$. The indicator function $\mathbb{I}(x, u)$ is approximated as

$$I(x) = 1 - \frac{1}{1 + e^{-v(r^2 - \|x - x_s\|_2^2 + \frac{1}{v})}},$$

so that it is smooth and the numerical solver can handle it. The parameter v and r are set as 10^8 and 10^{-6}, respectively. The coefficient α is set as 0.001 and the algorithm runs for 25 iterations. In Fig. 4, we show the total cost $\sum_{i=0}^{T_j}[(l(x_j(i), u_j(i)) - l(x_s, u_s))\delta + \alpha I(x_j(i))]$ improvement

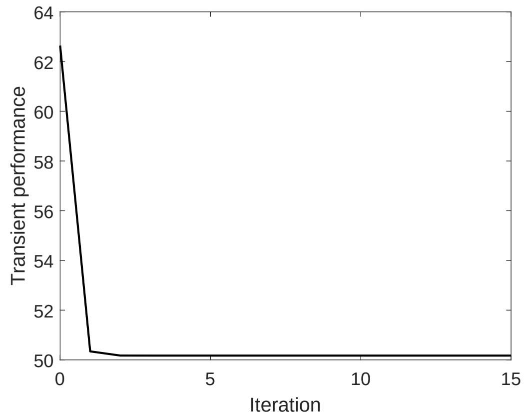

Fig. 1. Total cost improves along iterations

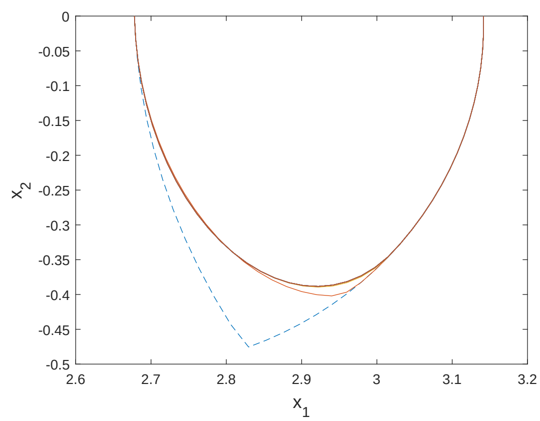

Fig. 2. State trajectory of each iteration (dash curve is the initial feasible state trajectory with $T = 39$)

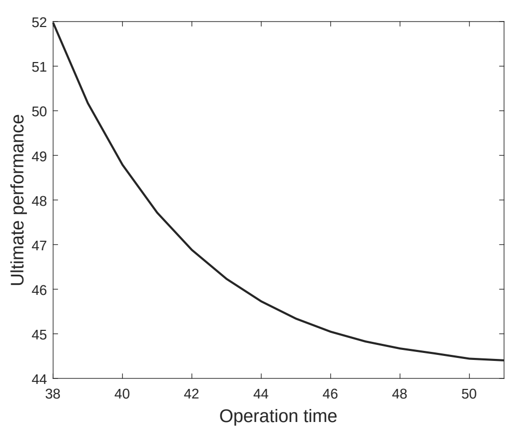

Fig. 3. Ultimate performance improves as operation time increases

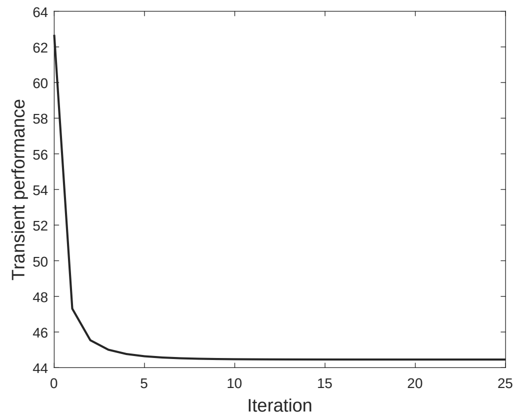

Fig. 4. Performance improves along iterations

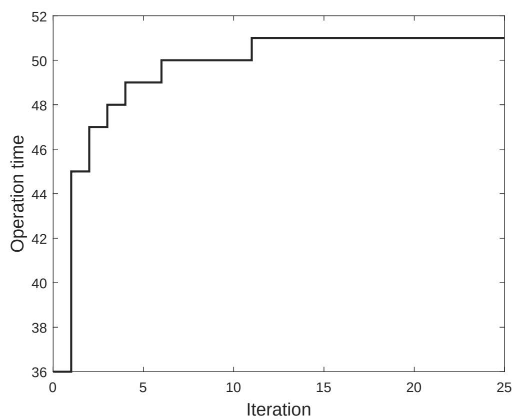

Fig. 5. Operation time is extended along iterations

along iterations. The ultimate performance $\sum_{i=0}^{T_j}(l(x_j(i), u_j(i)) - l(x_s, u_s))\delta$ with $j = 25$ is 44.4041, which is close to the limit performance shown in the previous example. In Fig. 5, we can observe that the algorithm extends the operation time along iteration until $T = 51$. The state trajectory of each iteration is shown in Fig. 6.

The next example is to show that if the operation time of the initial trajectory is too long, the proposed algorithm can also shorten it so that performance can be improved. We start from an initial trajectory reaching the equilibrium at $k = 60$. In Fig. 7, we show the total cost $\sum_{i=0}^{T_j}[(l(x_j(i), u_j(i)) - l(x_s, u_s))\delta + \alpha I(x_j(i))]$ improvement along iterations. The ultimate performance $\sum_{i=0}^{T_j}(l(x_j(i), u_j(i)) - l(x_s, u_s))\delta$ with $j = 40$ is 44.4041, which is the same as the limit performance shown in the previous example. In Fig. 8, we can observe that the algorithm

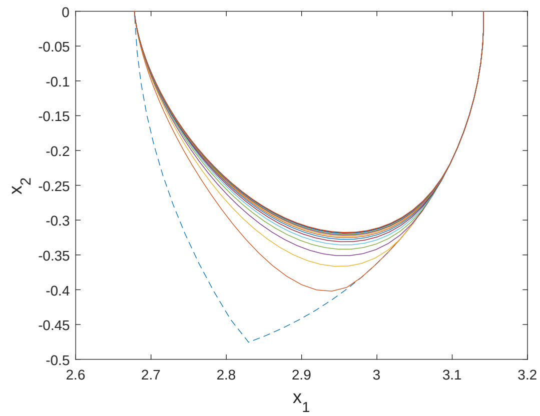

Fig. 6. State trajectory of each iteration (dash curve is the initial feasible state trajectory with $T = 36$)

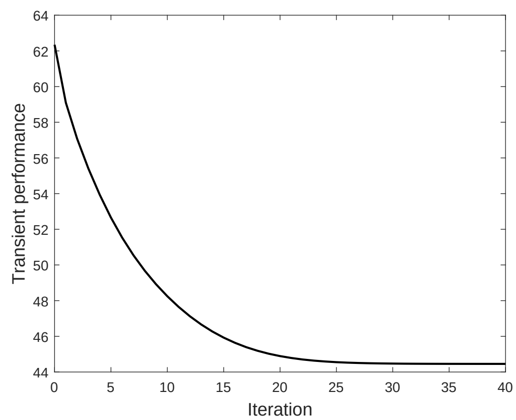

Fig. 7. Total cost improves along iterations

adjusts the operation time along iterations until $T = 51$. The state trajectory of each iteration is shown in Fig. 9.

V. Conclusion

In this paper, two learning-based economic model predictive control algorithms for iterative tasks have been proposed. The main features of the proposed control algorithms are: 1) it is capable of exploiting information from the last execution to improve the closed-loop performance; 2) the interested performance index is not limited to tracking error and can be general economic cost of the plant operation. We have proved that at each iteration, the performance index to be optimized will be no worse than that of the previous iteration. We also have shown that the two

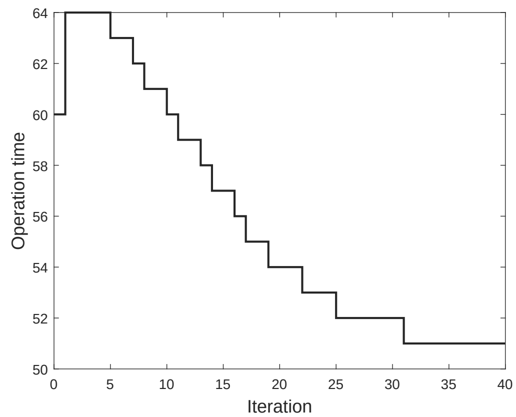

Fig. 8. Operation time is adjusted along iterations

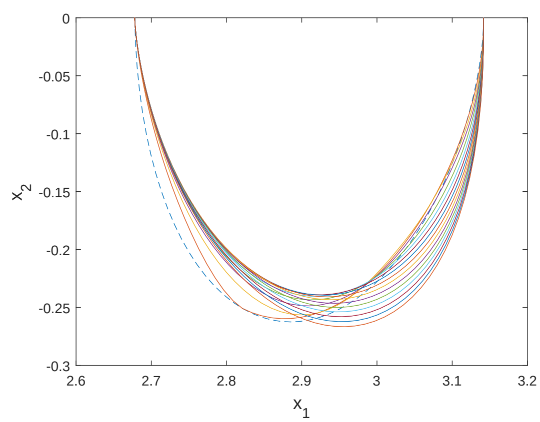

Fig. 9. State trajectory of each iteration (dash curve is the initial feasible state trajectory with $T = 60$)

proposed algorithms have uniformly upper bounded operation time so that each iteration can be terminated in finite time. The effectiveness of the proposed algorithms has been verified through two numerical examples.

VI. Acknowledgment

The work of Long and Xie is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

The work of Liu is funded by National Natural Science Foundation of China (NSFC) under Grants 61633014, 61733010 and Natural Science Foundation of Shandong Province, China (Grant No. ZR2018MF021).

REFERENCES

[1] A. Afram and F. Janabi-Sharifi, “Theory and applications of HVAC control systems – a review of model predictive control (MPC),” *Building & Environment*, vol. 72, pp. 343 – 355, 2014.

[2] D. Angeli, R. Amrit, and J. B. Rawlings, “On average performance and stability of economic model predictive control,” *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1615–1626, 2012.

[3] 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.

[4] Y. Chen and C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications*. Springer-Verlag London, 1999.

[5] 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.

[6] M. Diehl, R. Amrit, and J. B. Rawlings, “A Lyapunov function for economic optimizing model predictive control,” *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, March 2011.

[7] M. Ellis and P. D. Christofides, “On finite-time and infinite-time cost improvement of economic model predictive control for nonlinear systems,” *Automatica*, vol. 50, no. 10, pp. 2561 – 2569, 2014.

[8] L. Fagiano and A. R. Teel, “Generalized terminal state constraint for model predictive control,” *Automatica*, vol. 49, no. 9, pp. 2622 – 2631, 2013.

[9] P. Falugi, E. C. Kerrigan, and E. V. Wyk, *Imperial College London Optimal Control Software User Guide (ICLOCS)*. Department of Electrical Engineering, Imperial College London, London, UK., 2014.

[10] W. Langson, I. Chryssochoos, S. Raković, and D. Mayne, “Robust model predictive control using tubes,” *Automatica*, vol. 40, no. 1, pp. 125 – 133, 2004.

[11] J. H. Lee, K. S. Lee, and W. C. Kim, “Model-based iterative learning control with a quadratic criterion for time-varying linear systems,” *Automatica*, vol. 36, no. 5, pp. 641 – 657, 2000.

[12] 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, Oct 2000.

[13] 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.

[14] X. Li, Y.-H. Liu, and H. Yu, “Iterative learning impedance control for rehabilitation robots driven by series elastic actuators,” *Automatica*, vol. 90, pp. 1 – 7, 2018.

[15] X. Liu and X. Kong, “Nonlinear fuzzy model predictive iterative learning control for drum-type boiler–turbine system,” *Journal of Process Control*, vol. 23, no. 8, pp. 1023 – 1040, 2013.

[16] Y. Long, L. Xie, and S. Liu, “Iterative Learning Economic Model Predictive Control,” *ArXiv e-prints*, Jan. 2018.

[17] J. Ma, S. J. Qin, and T. Salsbury, “Application of economic MPC to the energy and demand minimization of a commercial building,” *Journal of Process Control*, vol. 24, no. 8, pp. 1282 – 1291, 2014.

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

[19] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.

[20] S.-K. Oh and J. M. Lee, “Iterative learning model predictive control for constrained multivariable control of batch processes,” *Computers & Chemical Engineering*, vol. 93, no. Supplement C, pp. 284 – 292, 2016.

[21] S. Qin and T. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[22] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, 2017.

[23] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks. a data-driven control framework,” *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2017.

[24] T. Samad, “A survey on industry impact and challenges thereof [technical activities],” *IEEE Control Systems*, vol. 37, no. 1, pp. 17–18, 2017.

[25] K. Tan, S. Huang, T. Lee, and A. Tay, “Disturbance compensation incorporated in predictive control system using a repetitive learning approach,” *System & Control Letters*, vol. 56, no. 1, pp. 75 – 82, 2007.

[26] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, “Model predictive control: A review of its applications in power electronics,” *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 16–31, March 2014.

[27] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[28] Y. Wang, V. Puig, and G. Cembrano, “Non-linear economic model predictive control of water distribution networks,” *Journal of Process Control*, vol. 56, pp. 23 – 34, 2017.

[29] S. Yu, C. Maier, H. Chen, and F. Allgöwer, “Tube MPC scheme based on robust control invariant set with application to lipschitz nonlinear systems,” *System & Control Letters*, vol. 62, no. 2, pp. 194 – 200, 2013.