Systems & Control Letters 94 (2016) 97–106

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Iterative learning control with input sharing for multi-agent consensus tracking

Shiping Yang [a,b], Jian-Xin Xu [b,*], Xuefang Li [b]

[a] Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore
[b] Department of Electrical and Computer Engineering, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 29 September 2015
Received in revised form
21 March 2016
Accepted 31 May 2016

Keywords:
Multi-agent system
Consensus tracking
Input sharing
Iteration-varying graph

ABSTRACT

In this paper, a novel iterative learning control (ILC) scheme with input sharing is presented for multi-agent consensus tracking. In many ILC works for multi-agent coordination problem, each agent maintains its own input learning, and the input signal is corrected by local measurements over iteration domain. If the agents are allowed to share their learned inputs among them, the strategy can improve the learning process as more learning resources are available. In this work, we develop a new type of learning controller by considering the input sharing among agents, which includes the traditional ILC strategy as a special case. The convergence condition is rigorously derived and analyzed as well. Furthermore, the proposed controller is extended to multi-agent systems under iteration-varying graph. It turns out that the developed controller is very robust to communication variations. In the numerical study, three illustrative examples are presented to show the effectiveness of the proposed controller. The learning controller with input sharing demonstrates not only faster convergence but also smooth transient performance.

1. Introduction

Iterative learning control (ILC), as an effective control strategy, is designed to improve the current performance of uncertain systems by fully utilizing the past control experience [1–6]. The control structure of ILC law is very straightforward. The current control signal is usually generated by the previous control signal plus some correction term, which may consist of previous or current tracking error. ILC is developed for control tasks that repeat in a fixed time interval, and requires only the system gradient bounds instead of accurate system model. Due to this appealing feature and its simplicity in implementation, ILC has been widely applied in practice, for example, X–Y table, robotic manipulators, chemical batch reactors, electric motors, etc.

In the past decades, multi-agent coordination problems have attracted heavy attention from the control community. In particular, the consensus problem [7–10] has been extensively investigated since many coordination problems can be formulated and solved under the framework of consensus, such as the multi-agent formation, swarming, rendezvous, search and rescue, coverage, distributed optimization, sensor fusion, etc. It is observed that many industry problems require repetitive executions and coordinations among several subsystems. For instance, from the operational point of view, it is very useful for a group of satellites orbiting the earth in formation for monitoring or positioning purpose [11]. As the satellite moving around the earth is a periodic task, the formation problem can fit perfectly in the ILC framework. Another example is the cooperative load transportation by several mobile robots [12,13]. Due to the task's repetitiveness, the cooperative transportation can also be studied in the ILC framework. These observations motivate the study of coordination problem by ILC. Comparing to the traditional control techniques, such as the sliding mode control [14] and neural network based adaptive control [15], there are a number of distinct features about ILC. First, ILC is designed to handle repetitive tasks. The traditional control methods cannot deal with or take advantage of the periodic nature. Under a repeatable control environment, repeating the same feedback would yield the same control performance. While by incorporating learning, ILC is able to improve the control performance iteratively. Second, the control objective is different. ILC tries to achieve perfect tracking, which means the tracking error is uniformly zero during the whole operation interval. Whereas, its counterparts usually achieve asymptotic error convergence in the time domain. Last but not least, ILC is one of the model-free control methods. As long as an appropriate learning gain is chosen, the perfect tracking can be achieved even when the system parameters are unknown.

* Corresponding author.
E-mail addresses: yangshiping001@gmail.com (S. Yang), elexujx@nus.edu.sg (J.-X. Xu), xuefangli@u.nus.edu (X. Li).

http://dx.doi.org/10.1016/j.sysconle.2016.05.017
0167-6911/

Recently, a number of ILC works for formation control and consensus tracking problems have been reported in the literature [16–24]. The distributed ILC laws in these works have a common structure, that is, each individual agent maintains its own learning process, and the correction term is synthesized by the local measurement (extended tracking error). Besides, they have no communication over their learned information. However, if the control inputs for some agents are already close to the desired control signals, these agents may help other agents by sharing their learned information. Therefore, all the agents in the system will be better off towards the global objective, i.e., reaching consensus. Based on this idea, a new type of learning controller is developed in this work. The new controller has two types of learning mechanisms. On one hand, each agent observes the behavior of the agents in their neighborhood, and constructs the correction term by its local measurement. This is the typical learning method in the majority of consensus literature. On the other hand, each agent shares its learned control input signal with their neighbors. As such, the two learning mechanisms are combined in hope of enhancing the learning process. The main contribution of this work is the incorporation of input sharing into the learning controller. The convergence condition of the proposed controller is rigorously derived and analyzed. To demonstrate the robustness of the proposed controller to the communication variations, ILC with input sharing is extended to multi-agent systems with iteration-varying graph. As the new learning controller combines two learning mechanisms, the traditional ILC renders a special case. This point is verified by the convergence condition. To demonstrate the performance of the new learning controller, three numerical examples are provided in the end. It shows that the new learning controller not only improves the convergence rate, but also smooth the transient performance.

This paper is organized as follows. In Section 2, basic terminologies in graph theory and λ-norm are firstly introduced. Next, the consensus tracking problem is formulated. Then, the controller design and convergence analysis are developed in Section 3. In Section 4, the proposed controller is extended to multi-agent systems with iteration-varying graph. To demonstrate the effectiveness of the results, two numerical examples are presented in Section 5. Lastly, Section 6 draws the conclusion.

2. Preliminaries and problem formulation

2.1. Preliminaries

Graph theory has been adopted in multi-agent coordination problem for many decades, and it will be used in this work to describe the communication among agents. Therefore, the basic terminologies in graph theory are briefly revisited below.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a weighted directed graph with the set of vertices $\mathcal{V} = \{1, 2, \ldots, N\}$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let $\mathcal{V}$ also be the index set representing the agents in the networked systems. A directed edge from k to j is denoted by an ordered pair $(k, j) \in \mathcal{E}$, which means that agent j can receive information from agent k. In this case, the vertex k is called the parent of j. The set of neighbors of the kth agent is denoted by $\mathcal{N}_k = \{j \in \mathcal{V} | (j, k) \in \mathcal{E}\}$. $\mathcal{A} = (a_{k,j}) \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix of $\mathcal{G}$ with nonnegative entries. In particular, $a_{k,k} = 0$, $a_{k,j} = 1$ if $(j, k) \in \mathcal{E}$, and $a_{k,j} = 0$ otherwise. The in-degree of vertex k is defined as $d_k^{in} = \sum_{j=1}^{N} a_{k,j}$, and the Laplacian of $\mathcal{G}$ is defined as $L = \mathcal{D} - \mathcal{A}$, where $\mathcal{D} = \mathrm{diag}(d_1^{in}, \ldots, d_N^{in})$. A spanning tree is a graph, and each vertex has exactly one parent except for one vertex which has no parent and is called the root. A graph is said to have or contain a spanning tree if the vertices set $\mathcal{V}$ and a subset of the edges set $\mathcal{E}$ can form a spanning tree.

Throughout this work, $|\cdot|$ is any generic vector norm, and the corresponding matrix norm is the induced matrix norm.

Next, we introduce the λ-norm, which is essentially an exponentially time weighted norm.

Definition 1. Given a vector function $\mathbf{f} : [0, T] \to \mathbb{R}^n$, its λ-norm is defined by

$$\|\mathbf{f}\|_\lambda = \max_{t\in[0,T]} e^{-\lambda t} |\mathbf{f}(t)|,$$

where λ is a positive constant.

In the ILC convergence analysis, λ-norm can be used to suppress the effects of system dynamics and reveal the input output relations directly, which makes the convergence proof simpler.

2.2. Problem formulation

Consider a group of N homogeneous dynamic agents, and the jth agent is governed by the following linear time-invariant model,

$$\begin{cases} \dot{\mathbf{x}}_{i,j}(t) = A\mathbf{x}_{i,j}(t) + B\mathbf{u}_{i,j}(t) \\ \mathbf{y}_{i,j}(t) = C\mathbf{x}_{i,j}(t) \end{cases} \quad \forall j \in \mathcal{V}, \tag{1}$$

where i denotes the iteration number, $\mathbf{x}_{i,j} \in \mathbb{R}^n$ is the state vector, $\mathbf{y}_{i,j} \in \mathbb{R}^p$ is the output vector, $\mathbf{u}_{i,j} \in \mathbb{R}^m$ is the control input, and A, B, C are constant matrices of compatible dimensions. For simplicity, the time argument, t, is dropped when no confusion arises.

The desired consensus trajectory, or the (virtual) leader's trajectory, is $\mathbf{y}_d(t)$ defined on a finite-time interval $[0, T]$, which is generated by the following dynamics,

$$\begin{cases} \dot{\mathbf{x}}_d = A\mathbf{x}_d + B\mathbf{u}_d, \\ \mathbf{y}_d = C\mathbf{x}_d, \end{cases} \tag{2}$$

where $\mathbf{u}_d$ is the continuous and unique desired control input.

Due to communication or sensor limitations, the leader's trajectory is only accessible to a small portion of the followers. Let the communication among followers be described by the graph $\mathcal{G}$. If the leader is labeled by vertex 0, then the complete information flow among all the agents can be characterized by a new graph $\overline{\mathcal{G}} = \{0 \cup \mathcal{V}, \overline{\mathcal{E}}\}$, where $\overline{\mathcal{E}}$ is the new edge set.

Let the tracking error for the jth agent at the ith iteration be $\mathbf{e}_{i,j} = \mathbf{y}_d - \mathbf{y}_{i,j}$. Assuming the tracking task is repeatable, the control objective is to design a set of distributed learning controllers such that the tracking error converges to zero along the iteration axis, i.e., $\lim_{i\to\infty} \|\mathbf{e}_{i,j}\| = 0$ for $j \in \mathcal{V}$, where $\|\mathbf{e}_{i,j}\|$ denotes the supremum norm and its defined as $\|\mathbf{e}_{i,j}\| = \max_{t\in[0,T]} |\mathbf{e}_{i,j}(t)|$.

To simplify the controller design and convergence analysis, the following two assumptions are imposed.

Assumption 1. *CB* is full column rank.

Remark 1. Assumption 1 implies that the relative degree of system (1) is well defined and it is exactly 1. When CB is not full rank, high-order derivative of the tracking error can be utilized in the controller design, and perfect consensus tracking can still be achieved.

Assumption 2. The initial state of all agents are reset to the desired initial state at every iteration, i.e., $\mathbf{x}_{i,j}(0) = \mathbf{x}_d(0)$.

Remark 2. Assumption 2 is the well-known identical initialization condition (i.i.c.), which is one of the fundamental problems or postulates in the ILC literature [2]. It has been used in many multi-agent coordination problems, for example the formation problems considered in [16,17]. To remove this condition requires either extra system information or additional control mechanisms. For example, if the initial state is manipulatable and some of system