Systems & Control Letters 69 (2014) 80–89

Contents lists available at ScienceDirect

Systems & Control Letters

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

Optimal iterative learning control design for multi-agent systems consensus tracking

Shiping Yang [a,b,*], Jian-Xin Xu [b], Deqing Huang [c], Ying Tan [d]

[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*
[c] *Department of Aeronautics, Imperial College London, United Kingdom*
[d] *Department of Electrical and Electronic Engineering, University of Melbourne, Australia*

ARTICLE INFO

Article history:
Received 29 April 2013
Received in revised form
1 October 2013
Accepted 2 April 2014
Available online 22 May 2014

Keywords:
Iterative learning control
Consensus tracking
Multi-agent systems
Optimal design

ABSTRACT

Under a repeatable operation environment, this paper proposes an iterative learning control scheme that can be applied to multi-agent systems to perform consensus tracking under the fixed communication topology. The agent dynamics are modeled by time-varying nonlinear equations which satisfy the global Lipschitz continuous condition. In addition, the desired consensus trajectory is only accessible to a subset of the followers. By using the concept of the graph dependent matrix norm, the convergence conditions can be specified at the agent level, which depend on a set of eigenvalues that are associated with the communication topology. The results are first derived for homogeneous agent systems and then extended to heterogeneous systems. Next, optimal controller gain design methods are proposed in the sense that the λ-norm of tracking error converges at the fastest rate, which imposes a tightest bounding function for the actual tracking error in the λ-norm analysis framework. In the end, an illustrative example of a group of heterogeneous agents is provided to demonstrate the effectiveness of the proposed design methods.

1. Introduction

In the past decade, multi-agent systems have attracted considerable attention from many researchers of various backgrounds due to their potential applications and cross-disciplinary nature. *Consensus* is an important class of multi-agent systems coordination problems. According to [1], in networks of agents (or dynamic systems), consensus means to reach an agreement regarding certain quantities of interest that are associated with the agents. In a consensus realization, the control action of an agent only depends on the information received or measured from its neighborhood. Since the control law is a kind of distributed algorithm, it is more robust and scalable compared to centralized control algorithms.

Consensus algorithm is a very simple local coordination rule which can result in very complex and useful behaviors at the group level. For instance, it is widely observed that by adopting such a strategy, a school of fish can improve the chance of survival under the sea [2]. Many interesting coordination problems have been formulated and solved under the framework of consensus, e.g., distributed sensor fusion [1], satellite alignment problem [3], multi-agent formation [4], synchronization of coupled oscillators [5], and optimal dispatch in power systems [6]. To solve a consensus problem, the communication among agents is one of the indispensable components. In the early stage, the communication graph is assumed to be fixed. However, a consensus algorithm that is insensitive to topology variations, is more desired since many practical conditions can be modeled as time-varying communication, for example, asynchronous updating, communication link failures and creations. By using the set-valued Lyapunov theory, consensus over time-varying communication is investigated in [7]. In [8–10], consensus over random networks or stochastically switching graph is considered. It is shown the almost sure consensus can be achieved in such kind of probabilistic settings. Many standard communication assumptions used in the literature and consensus results are summarized in [11].

Iterative learning control (ILC) is a matured intelligent learning control strategy, which fully utilizes the past control experience to improve the tracking performance in the current iteration. The current control signal is usually synthesized by the past control signals and correction terms which consist of tracking errors in the

* Corresponding author at: Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore.
E-mail addresses: yangshiping@nus.edu.sg (S. Yang), elexujx@nus.edu.sg (J.-X. Xu), elehd2012@gmail.com (D. Huang), yingt@unimelb.edu.au (Y. Tan).

http://dx.doi.org/10.1016/j.sysconle.2014.04.009
0167-6911/

past and/or current iterations. It mimics the human learning process that skills can be improved by repetition. ILC is developed for control tasks that repeat in a fixed and finite-time interval, and requires only the system gradient bound instead of accurate system model. Most practical plants only have nominal models instead of the accurate models, that makes ILC have great potential for applications in practice. For example, X–Y table, chemical batch reactors, robotic manipulators. ILC was initially proposed in 1984 [12], and has been widely explored since then [13–16].

The main motivation of this work is to address many practical coordination problems that are repeatable or require periodic executions, such as the trajectory keeping in satellite formation problem considered in [17]. Since the satellites orbiting the earth is a periodic task, from an operational point of view, it is useful for a group of satellites maneuvering in certain shape of formation. Thus, this problem can be well formulated in the framework of ILC. The idea of using ILC for multi-agent coordination first appears in [18], where multi-agent formation control problem is studied for a group of global Lipschitz nonlinear systems, in which the communication graph is identical to the formation structure. When the tree-like formation is considered, the perfect formation control can be achieved. In [19], by incorporating with high-order internal model ILC, an iteratively switching formation problem is formulated and solved in the same framework. The communication graphs are supposed to be direct spanning trees as well. Ref. [20] is dedicated to improving the control performance in [17]. The formation structure can be independent of the communication topology, and time-varying communication is assumed in [20]. The convergence condition is specified at the group level by a matrix norm inequality, and the learning gain can be designed by solving a set of linear matrix inequalities (LMIs). It is not clear under what condition the set of LMIs admits a solution, and it is the lack of insight how the communication topologies relate to the convergence condition. In [21], the idea of terminal ILC [22] is brought into the consensus problem. A finite-time consensus problem is formulated for discrete-time linear systems in ILC framework. It is shown that all the agents reach consensus at the terminal time as iteration number goes to infinity. In [23], the authors extend the terminal consensus problem in their previous work to track a time-varying reference trajectory over the entire finite-time interval. A unified ILC algorithm is developed for both discrete-time and continuous-time linear agents. Necessary and sufficient conditions in the form of spectral radius are derived to ensure the convergence properties. As mentioned before, communication is very important in multi-agent coordination. Fixed communication is assumed in all the above mentioned ILC works except [20], in which the communication is a connected graph at every time instance. A preliminary analysis of ILC algorithm over iteration-varying communication is reported in [24]. It shows that consensus tracking can be achieved if the union graph over a fixed number of iterations contains a spanning tree.

In this work, we study the consensus tracking problem for a group of time-varying nonlinear dynamic agents, where the nonlinear terms satisfy the global Lipschitz continuous condition. The communication graph is assumed to be fixed. In comparison with the current literature, the main challenges and contributions of this work are summarized below: (1) in [23], the convergence condition for continuous-time agents is derived based on the result in 2-dimensional system theory [25, Theorem 1], which is only valid for linear systems. By adoption of a graph dependent matrix norm and λ-norm analysis, we are able to obtain the results for global Lipschitz nonlinear systems; (2) in [20], the convergence condition is specified at the group level in the form of a matrix norm inequality, and learning gain is designed by solving a set of LMIs. Nevertheless, owing to the graph dependent matrix norm, the convergence condition is expressed at the individual agent level in the form of spectral radius inequalities in our work, which are related to the eigenvalues associated with the communication graph. It shows that these eigenvalues play crucial roles in the convergence condition. In addition, the results are more general than the matrix norm inequality since the spectral radius of a matrix is less or equal to its matrix norm; (3) by using the graph dependent matrix norm and λ-norm analysis, the learning controller design can be extended to heterogeneous systems. However, the methods in [20,23] are not directly applicable to heterogeneous systems; (4) the obtained convergence condition motivates us to consider optimal learning gain designs which can impose the tightest bounding functions for the actual tracking errors. A preliminary version of this work is published in [26].

This paper is organized as follows. In Section 2, notations, basic terminologies in algebraic graph theory, and some useful results are introduced. Next, the consensus tracking problem for heterogeneous agents is formulated. Then, ILC control laws are developed in Section 3, for both homogeneous and heterogeneous agents, where the convergence condition is analyzed rigorously. Next, optimal learning design methods are proposed in Section 4, where optimal designs for undirected and directed graphs are explored respectively. Then, an illustrative example for heterogeneous agents under fixed directed graph is given in Section 5 to demonstrate the efficacy of the proposed algorithms. Finally, we conclude the paper in Section 6.

2. Preliminaries and problem formulation

2.1. Preliminaries

The set of real numbers is denoted by $\mathbb{R}$, and the set of complex numbers is denoted by $\mathbb{Z}$. The set of integers is denoted as $\mathbb{N}$, and $i \in \mathbb{N}_{\geq 0}$ is the number of iteration. For any $z \in \mathbb{Z}$, $\Re(z)$ denotes its real part. For a given vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$, $|\mathbf{x}|$ denotes any l_p vector norm, where $1 \leq p \leq \infty$. In particular, $|\mathbf{x}|_1 = \sum_{k=1}^{n} |x_k|$, $|\mathbf{x}|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$, and $|\mathbf{x}|_\infty = \max_{k=1,\dots,n} |x_k|$. For any matrix $A \in \mathbb{R}^{n\times n}$, $|A|$ is the induced matrix norm. $\rho(A)$ is its spectral radius. Moreover, $\otimes$ denotes the Kronecker product, and I_m is the $m \times m$ identity matrix.

Let $\mathcal{C}^m[0, T]$ denote a set consisting of all functions whose mth derivatives are continuous on the finite-time interval $[0, T]$. For any function $\mathbf{f}(\cdot) \in \mathcal{C}^m[0, T]$, the supremum norm is defined as $\|\mathbf{f}\| = \max_{t\in[0,T]} |\mathbf{f}(t)|$. Let λ be a positive constant, the time weighted norm (λ-norm) is defined as $\|\mathbf{f}\|_\lambda = \max_{t\in[0,T]} e^{-\lambda t}|\mathbf{f}(t)|$.

Graph theory is an instrumental tool to describe the communication topology in the multi-agent systems, the basic terminologies in graph theory are briefly introduced below.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ be a weighted directed graph with the set of vertices $\mathcal{V} = \{1, 2, \dots, N\}$, the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and the adjacency matrix $\mathcal{A}$. Let $\mathcal{V}$ also be the index set representing the agents in the systems. A direct 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, k is called the parent of j, and j is the child of k. The neighborhood of the kth agent is denoted by the set $N_k = \{j \in \mathcal{V} | (j, k) \in \mathcal{E}\}$. $\mathcal{A} = (a_{k,j}) \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix of $\mathcal{G}$. In particular, $a_{k,k} = 0$, $a_{k,j} = 1$ if $(j, k) \in \mathcal{E}$, and $a_{k,j} = 0$ otherwise.[1] 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 $\mathcal{D} = \mathrm{diag}(d_1^{in}, d_2^{in}, \dots, d_N^{in})$. A spanning tree is a directed graph, whose vertices have exactly one parent except for one vertex, which is called the root who has no parent. We say that a graph contains or has a spanning tree if $\mathcal{V}$ and a subset of $\mathcal{E}$ can form a spanning tree.

[1] Undirected graph is a special case of directed graph, satisfying $a_{k,j} = a_{j,k}$.