Automatica 69 (2016) 258–264

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Iterative learning control with time-partitioned update for collaborative output tracking[☆]

Santosh Devasia

Mechanical Engineering Department, University of Washington, Seattle, 98195-2600, USA

ARTICLE INFO

Article history:
Received 3 November 2014
Received in revised form
13 October 2015
Accepted 9 February 2016

Keywords:
Iterative learning control
Convergence
Cooperative control
Heterogeneous systems
Output tracking

ABSTRACT

This article studies iterative learning control (ILC) where multiple heterogeneous linear subsystems (with potentially different individual dynamics) update their input simultaneously based on the error in a collaboratively controlled desired output. A challenge is that convergence of iterative learning for each individual subsystem (when the other subsystems are not learning) may not guarantee convergence under collaborative-(co-) learning. This work proposes an update-partitioning approach for co-learning and demonstrates convergence whenever the individual, iterative learning for each subsystem is convergent. The main contribution of this work is to show that any unity partition (where the sum of the partition is one) of the update law ensures convergence of the co-learning. Since the time partitioning of the update can be chosen independent of the individual learning (convergence) rate of the subsystems, the proposed approach enables the separate design of each individual subsystem's input-update law followed by conjoining of the individual update laws for co-learning using the partitioning approach. Additionally, an intermittent time partitioning is developed when the desired trajectory is not known to all (but only some) of the co-learning subsystems.

1. Introduction

This article studies iterative learning control (ILC) where multiple (n) heterogeneous linear subsystems (with potentially different individual dynamics, G_i) update their input u_i simultaneously based on the error in a collaboratively controlled output y described in the Laplace domain by $y(s) = \sum_{i=1}^{n} G_i(s)u_i(s)$. Such collaborative control of a common output y arises in applications such as: (i) active orthosis where a joint is being controlled by a human and a machine, e.g., Bruce Wiggin, Sawicki, and Collins (2011); (ii) dual-stage positioners where a high-bandwidth low-range actuator augments the performance of a low-bandwidth but large-range actuator, e.g., Schroeck, Messner, and McNab (2001) and Brinkerhoff and Devasia (2000). Multiple (redundant) actuators have also been used for collaborative positioning, e.g., using independent actuators or legs to move a single stage, e.g., Kim and Chang (2013) and Wilcox and Devasia (2015), which is similar to use of multiple actuators in nature such as arrays of cilia for movement in fluid environments, multiple muscle fibers in biological actuators, and multiple-legs in centipedes and millipedes. With increasingly distributed sensing, computing and actuation, there is interest in each of the subsystems to operate somewhat independently, which can enable the potential recruitment of different actuator subsystems for varying tasks, e.g., similar to the recruitment of different muscle fiber sets (motor units) to manage changing muscle loads. Moreover, since complex tasks could be broken into previously designed trajectories, there is interest in learning to control specific desired (pre-specified) trajectories. Therefore, this work investigates the iterative learning of a common desired output trajectory with multiple subsystems.

Convergence of iterative learning to achieve output tracking of a single output, with a single subsystem, has been well studied, e.g., Arimoto, Kawamura, and Miyazaki (1984), Ghosh and Paden (2002), Tien, Zou, and Devasia (2005), Ahn, Chen, and Moore (2007), Ghosh and Paden (2001), Mishra and Tomizuka (2005) and Bristow and Alleyne (2008). Note that, for perfect control of a desired output $y = y_d$ with a single system G, the input u depends on the inverse G^{-1} of the system G, i.e., $u = G^{-1}y_d$. This has motivated the use of the inverse $\hat{G}^{-1}$ of the known model $\hat{G}$ of the system G in the iteration update law (Atkeson & McIntyre, 1986; Ghosh & Paden, 2002; Tien et al., 2005). For such cases, convergence to the desired output can be shown provided the phase error in the model

☆ This work was partially supported by the National Science Foundation (NSF) through grants CMII 1301452 and CMII 1536306. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Tong Zhou under the direction of Editor Richard Middleton.

E-mail address: devasia@uw.edu.

http://dx.doi.org/10.1016/j.automatica.2016.02.027
0005-1098/© 2016 Elsevier Ltd. All rights reserved.

is less than $\pi/2$ and the iteration gain is sufficiently small (Tien et al., 2005). This work aims to generalize such convergence conditions, developed for single input single output systems, to multiple subsystems that collaboratively learn to track a single desired output. However, for iterative co-learning with multiple subsystems, the output error depends on all the n subsystems $\{G_i\}_{i=1}^n$, and therefore the update for the input u_i for each subsystem (and therefore, the convergence condition) becomes dependent on the dynamics of the other subsystems. This interdependence implies that it can be challenging to determine convergence without knowledge of the dynamics of all the subsystems.

The specific issue addressed here is the question whether convergence of iterative learning for each individual subsystem (when the other subsystems are not learning) can be used to develop convergence conditions that do not require knowledge of the dynamics of all the subsystems. Note that convergence properties for special cases are known. For example, convergence follows if only one of the subsystems is actively learning and the others are not. Moreover, since typical convergence conditions constrain the gain of the update law to be sufficiently small, it is also possible to demonstrate that lack of co-ordination (e.g., when all the systems aim to learn simultaneously based on individual update laws) can lead to loss of convergence. For example, the gain of the update law increases and can lead to loss of convergence with increasing number n of (similar) subsystems that are using the same update law. Establishing convergence is also possible when the desired output y_d can be mapped into relatively independent predetermined desired output $y_{d,i}$ for each subsystem, e.g., for manipulation with multiple (but minimal number of) robotic arms (Kawasaki, Ueki, & Ito, 2006). Moreover, convergence can be established for consensus-type algorithms that seek to achieve the same output $y_i = y_d$ for each agent in networked multi-agent systems, e.g., Chen, Hua, and Ge (2014) and Choi, Oh, and Horowitz (2009)—in contrast, the current work aims to collaboratively control a single total output, $y_d = \sum_i y_i$, where all the individual subsystem outputs y_i are not required to be equal.

The main contribution of this work is to show that any unity partition (where the sum of the partition is one) of the update ensures convergence of the co-learning. While the selection of the update partitioning does require some centralization, it could also be as simple as an equal distribution of the update authority to each subsystem, i.e., a scaling of the update by $1/n$ where n is the number of subsystems, as shown with the simulation example. The specific partition can be frequency dependent and time varying, and therefore, be used to develop algorithms that modify the distribution of control effort among the different subsystems and potentially manage actuator bandwidth constraints in different frequency ranges as in Devasia (2002). Additionally, an intermittent time-partitioning-based update law is developed for the case when the desired trajectory is not known to all of the subsystems that are co-learning.

2. Problem formulation

The goal is to achieve desired output $y = y_d$ for a system with n linear subsystems $\{G_i\}_{i=1}^n$, described by

$$y(s) = \sum_{i=1}^{n} G_i(s)u_i(s). \tag{1}$$

The n inputs to achieve the desired output y_d are co-learnt in an iterative manner, with the update law described in the frequency domain at the kth iteration step,

$$u_{i,k+1}(\omega) = u_{i,k}(\omega) + \rho_{i,k}(\omega)\hat{G}_i^{-1}(\omega)\left[y_d(\omega) - y_k(\omega)\right] \tag{2}$$

for all frequency ω, $i = 1, \dots n$, where $\rho_{i,k}(\omega)$ is a real-valued scalar, k is a positive integer, y_k is the output,

$$y_k(\omega) = \sum_{i=1}^{n} G_i(\omega)u_{i,k}(\omega), \tag{3}$$

the term in the square bracket in Eq. (2) represents the output error, $\hat{G}_i$ is the known model of the ith subsystem G_i, and the value of a model G evaluated on the imaginary axis of the complex plane is defined as $G(\omega) = G(s)\Big|_{s=j\omega}$ with $j = \sqrt{-1}$.

Assumption 1 (*System and Model Properties*). In the following, each subsystem G_i ($i = 1, \dots, n$) and its model $\hat{G}_i$ are stable with hyperbolic zero dynamics, i.e., all zeros have a nonzero real part. Also, these transfer functions are not trivial $G_i \neq 0$ and $\hat{G}_i \neq 0$.

Remark 1 (*Model Invertibility*). Hyperbolic zero dynamics implies invertibility of the model $\hat{G}_i(\omega)$ in the input update law in Eq. (2), and also ensures robustness of the inverse under modeling uncertainty (Devasia, 2002).

With a single input u_m, the iterative control in Eq. (2) converges if the modeling error is sufficiently small, as shown in Tien et al. (2005), and stated formally below.

Lemma 1 (*Single-input Convergence*). *With only one input u_i, i.e., $u_{m,1} = 0$ and $\rho_{m,k} = 0$, for all $m \neq i$, a finite initial input $u_{i,1}(\omega)$ and a fixed iteration gain $\overline{\rho}_i(\omega) = \rho_{i,k}(\omega)$, the single-input iterations in Eq. (2) converges to the inverse input $u_{i,inv}$ at frequency ω, i.e., $\lim_{k\to\infty} u_{i,k}(\omega) = u_{i,inv}(\omega) = G_i^{-1}(\omega)\left[y_d(\omega)\right]$, which results in exact tracking of the desired output $y_d(\omega)$, i.e., $\lim_{k\to\infty} y_k(\omega) = \lim_{k\to\infty} G_i(\omega)u_{i,k}(\omega) = y_d(\omega)$ if and only if the magnitude of the phase uncertainty $\Delta_{\Phi,i}$ in the model and the update gain $\overline{\rho}_i(\omega)$ are sufficiently small*

$$\begin{aligned} &|\Delta_{\Phi,i}(\omega)| < \pi/2 \\ &0 < \overline{\rho}_i(\omega) < \frac{2\cos\left[\Delta_{\Phi,i}(\omega)\right]}{\Delta_{M,i}(\omega)} = \rho_i^*(\omega) \end{aligned} \tag{4}$$

where the magnitude uncertainty $\Delta_{M,i}$ and the phase uncertainty $\Delta_{\phi,i}$ are defined by

$$\frac{G_i(\omega)}{\hat{G}_i(\omega)} = \Delta_{M,i}(\omega)e^{j\Delta_{\phi,i}(\omega)}. \tag{5}$$

Proof. This follows from Lemma 1 in Tien et al. (2005). □

Assumption 2. In the following, it is assumed that the modeling uncertainty for each subsystem G_i and the associated fixed gain $\overline{\rho}_i(\omega) \neq 0$ satisfy the single-input convergence conditions in Eq. (4).

The challenge to establish convergence with co-learning is the interdependence of the subsystems in the update law. In particular, the input update can be rewritten from Eqs. (1), (2) as

$$\begin{aligned} &u_{i,k+1}(\omega) - u_{i,k}(\omega) \\ &= \left(1 - \frac{\rho_{i,k}(\omega)G_i(\omega)}{\hat{G}_i(\omega)}\right)\left[u_{i,k}(\omega) - u_{i,k-1}(\omega)\right] \\ &\quad - \sum_{j=1, j\neq i}^{n} \frac{\rho_{i,k}(\omega)G_j(\omega)}{\hat{G}_i(\omega)}\left[u_{j,k}(\omega) - u_{j,k-1}(\omega)\right]. \end{aligned}$$

Without the second term on the right hand side, the input-learning iterations would be a contraction if the gain term $1 - (\rho_{i,k}(\omega)G_i(\omega)/\hat{G}_i(\omega))$ is small. However, the interdependence of the subsystems due to the coupling between the model $\hat{G}_i$ and the other subsystems G_j, in the second term, makes it challenging to develop conditions for co-learning. Hence, the problem is to