Control Engineering Practice 64 (2017) 74–87

Contents lists available at ScienceDirect

Control Engineering Practice

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

A multi-objective iterative learning control approach for additive manufacturing applications☆

Ingyu Lim[a], David J. Hoelzle[b], Kira L. Barton[a,*]

[a] Department of Mechanical Engineering at the University of Michigan at Ann Arbor, MI 48109, USA
[b] Department of Mechanical and Aerospace Engineering at the Ohio State University, Columbus, OH 43210, USA

A R T I C L E I N F O

Article history:
Received 29 December 2015
Received in revised form
23 March 2017
Accepted 25 March 2017

Keywords:
Iterative learning control
Multiple objective
Spatial domain
Additive manufacturing
Electrohydrodynamic Jet Printing

A B S T R A C T

Iterative learning control (ILC) is a method for improving the performance of stable, repetitive systems. Standard ILC is constructed in the temporal domain, with performance improvements achieved through iterative updates to the control signal. Recent ILC research focuses on reformulating temporal ILC into the spatial domain, where 2D convolution accounts for spatial closeness. This work expands spatial ILC to include optimization of multiple performance metrics. Performance objectives are classified into primary, complementary, competing, and domain specific objectives. New robustness and convergence criteria are provided. Simulation results validate flexibility of the spatial framework on a high-fidelity additive manufacturing system model.

1. Introduction

Iterative Learning Control (ILC) is an adaptive control approach for improving the performance of repetitive systems. Standard ILC algorithms use the error signal from previous iterations to generate an updated control signal for the current iteration to improve system performance Barton & Alleyne (2011); Bristow et al. (2006); Moore et al. (2006). ILC has been applied to a broad range of systems from manufacturing to chemical processes Barton & Alleyne (2008); Bristow & Alleyne,; Freeman et al. (2010); Lee & Lee (2007); Longman (2012). While ILC algorithms have conventionally been derived in the temporal domain, recent advancements have extended these algorithms to the spatial framework to apply ILC to systems for which the spatial dynamics play a particularly important role in determining system behavior Hoelzle & Barton (2016); Bristow & Alleyne (2006); Cichy et al. (2012); Moore et al. (2007); Sahoo et al. (2007). In these examples, ILC was used to improve system performance through updates to a performance metric defined in either the temporal domain Barton & Alleyne (2008); Barton & Alleyne (2011); Bristow & Alleyne,; Bristow et al. (2006); Freeman et al. (2010); Lee & Lee (2007); Longman (2012); Moore et al. (2006) or the spatial domain Hoelzle & Barton (2016); Bristow & Alleyne (2006); Cichy et al. (2012); Moore et al. (2007); Sahoo et al. (2007). Previous work by the authors in Hoelzle and Barton (2016) introduced the first formalized concept of a spatial iterative learning control (SILC) framework that was derived entirely in the spatial domain, with no direct or indirect connection to temporal dynamics. The work in Hoelzle and Barton (2016) outlined the 2D convolution-based structure for the SILC architecture, as well as a design procedure and convergence analysis for 'lifted' norm optimal and frequency domain controller designs.

Most applications of ILC focus on improving the performance of a single performance objective such as improving trajectory tracking. As systems become more complex and include integrated subsystems, the controller design must be governed by multiple performance objectives that align with the key performance metrics of the varying subsystems. Examples of such applications can be found in unmanned air vehicle surveillance Barton & Kingston (2013); Lim & Barton, (2014, 2013) (objectives: speed, path following, energy consumption, sensor transmission strength) and manufacturing Barton, Hoelzle, Alleyne, and Johnson (2011) (objectives: throughput, part quality, material usage, energy usage). Designing the controller to address two or more performance objectives provides a greater degree of performance flexibility.

To enable the learning algorithm to extend to a multi-objective controller design, a more flexible control structure must be defined such that underutilized control can be applied towards

☆This work was supported in part by NSF Awards CMMI-1434693 and CMMI-1434660, as well as start-up funds from the University of Michigan.

* Corresponding author.
E-mail addresses: ingyulim@umich.edu (I. Lim), hoelzle.1@osu.edu (D.J. Hoelzle), bartonkl@umich.edu (K.L. Barton).

http://dx.doi.org/10.1016/j.conengprac.2017.03.011
0967-0661/

multiple performance metrics. To achieve this control flexibility, the primary objective (e.g., typically a tracking requirement) is modified to focus on a select group of discrete states, $\chi(n_p) \subseteq g_d(k)$, where n_p are the selected points for all $p = 1, \ldots, \bar{M}$, $g_d(k)$ defines the desired performance profile, and k is the time index. The concept of point-based rather than trajectory-based learning control was first introduced in Ding and Wu (2007); Freeman, Cai, Rogers, and Lewin (2011); Park, Chang, Park, and Lee (2006); Van de Wijdeven and Bosgra (2008), in which the authors defined a point-to-point learning controller that focused on the performance at a select group of time points or coordinates. This point-based learning controller has since been applied towards applications in robotic pick n′ place tasks Dijkstra et al., , patient stroke rehabilitation Freeman et al. (2009), and reconnaissance missions with unmanned aerial vehicles Lim and Bang (2010). Once the primary objective has been defined, additional performance metrics can be identified. In the context of this research, we define three additional performance categories: complementary metrics that contribute to the overall weighting on the primary objective, and two classes of competing metrics that are inversely related to the primary metrics: one class that defines the metrics as a function of the output, and a second class that defines the metrics as a function of the input. Fig. 1 provides an illustrative schematic of the trade-off relationship between competing and primary objectives.

Recently published work in the area of multi-objective learning Freeman (2012); Freeman and Tan (2013); Owens, Freeman, and Chu (2013) utilizes a two-step approach to optimizing system performance. Step 1 aims at optimizing the control effort to achieve zero steady-state trajectory tracking (the primary objective in these applications). In step 2, the framework seeks to optimize the performance of an additional objective through the use of a cost function that considers the additional objective, while simultaneously minimizing the difference between a new control input and the optimal control signal determined in step 1. This iterative learning sequence involves multiple steps, while bounding the range of the new solution to be arbitrarily close to the initial optimal input.

Previous work by the authors in Lim & Barton (2014); , introduced a multi-objective learning control framework that optimizes multiple performance objectives simultaneously. As a result of our one-step optimization approach, the relationship between the primary objective and additional objectives can be clearly observed. Additionally, by eliminating the constraint on the

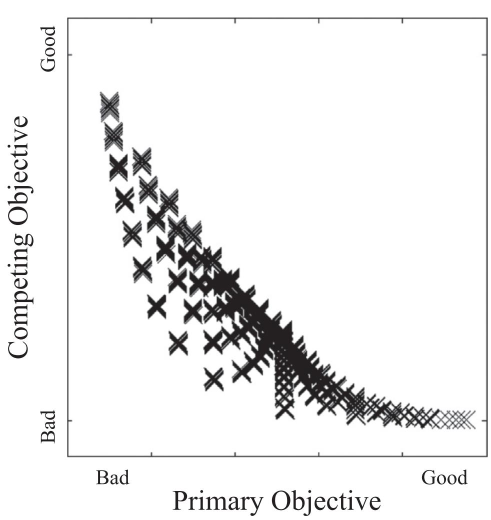

Fig. 1. Pareto Analysis: Example trade-off between two key performance objectives: primary versus a competing classification.

control signal that is imposed in the two step approach, the optimization search is implemented over a broader set of potential solutions. This enables a greater variety of possible outcomes. In this work, the authors extend the research from Hoelzle & Barton (2016); Lim & Barton (2014); , in two key areas:

1. The publication in Hoelzle and Barton (2016) formalized the idea of SILC and contained a demonstrative example. The manuscript herein expands on the SILC formalization in Hoelzle and Barton (2016), extending the single-objective (e.g. error minimization) cost function to a multi-objective optimization with full stability, convergence, and robustness analysis. The framework is validated through a simulation study from an empirically derived model. The 2D controller structure overlaps with Hoelzle and Barton (2016). However, the formalized multi-objective SILC framework, along with the specific stability and convergence analysis are unique contributions in this manuscript.
2. Our previous work in Lim and Barton (2014) set the ground work for the construction of a multi-objective framework. This work expands on the previous demonstrations through the formal definition of three classifications of additional performance metrics: complementary, competitive, and domain specific with robust performance and stability analysis provided for all three classification categories. While the additive manufacturing application was presented in ,, the formal definition of performance objectives specific to AM applications is a unique contribution.

The remainder of the paper is organized as follows. Section 2 introduces preliminary information used in the paper. The multi-objective iterative learning control framework is presented in Section 3. Controller design is given in Section 4. Section 5 provides the simulation background and results. Concluding remarks are given in Section 6.

2. Preliminaries

2.1. Spatial convolution

In this manuscript, the authors investigate multi-objective iterative learning control in the spatial domain. The authors first introduced the concept of a spatial iterative learning controller in Hoelzle & Barton,. A complete description of spatial ILC can be found in their more recent work in Hoelzle and Barton (2016). Here we present only the salient details from Hoelzle and Barton (2016) necessary for the introduction of a multi-objective learning controller in the spatial domain. For additional details and a more thorough discussion of the spatial learning framework, the readers are invited to see Hoelzle & Barton (2016).

A 2D spatial input map $u(x, y)$ and a plant operator H yield a spatial output map $g(x, y)$,

$$g(x, y) = Hu(x, y), \tag{1}$$

where $u(x, y) \in \mathbb{R}^{A\times B}$, $g(x, y) \in \mathbb{R}^{A\times B}$, and x and y are integer valued coordinates that discretize the spatial map; $x = 0, 1, \ldots, A - 1$ and $y = 0, 1, \ldots, B - 1$. The plant operator H is represented by a spatial impulse response centered around a point (m, n) Gonzalez & Wintz, 1977:

$$H\delta(x - m, y - n) = h(x - m, y - n),$$

where $h(x - m, y - n) \in \mathbb{R}^{C\times D}$. Assuming a spatially-invariant operator H, Eq. (1) can be computed using the 2D convolution sum,