Control Engineering Practice 18 (2010) 339–348

Contents lists available at ScienceDirect

Control Engineering Practice

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

Experimentally supported 2D systems based iterative learning control law design for error convergence and performance ☆

Lukasz Hladowski [a], Krzysztof Galkowski [a], Zhonglun Cai [b], Eric Rogers [b,*], Chris T. Freeman [b], Paul L. Lewin [b]

[a] Institute of Control and Computation Engineering, University of Zielona Gora, Podgorna 50, 65-246 Zielona Gora, Poland
[b] School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

ARTICLE INFO

Article history:
Received 4 May 2009
Accepted 2 December 2009
Available online 12 February 2010

Keywords:
Iterative learning control
LMIs
Robustness

ABSTRACT

This paper considers iterative learning control law design for both trial-to-trial error convergence and along the trial performance. It is shown how a class of control laws can be designed using the theory of linear repetitive processes for this problem where the computations are in terms of linear matrix inequalities (LMIs). It is also shown how this setting extends to allow the design of robust control laws in the presence of uncertainty in the dynamics produced along the trials. Results from the experimental application of these laws on a gantry robot performing a pick and place operation are also given.

1. Introduction

Iterative learning control (ILC) is a technique for controlling systems operating in a repetitive (or pass-to-pass) mode with the requirement that a reference trajectory $y_{ref}(t)$ defined over a finite interval $0 \le t \le \alpha$, where α denotes the trial length, is followed to a high precision. Examples of such systems include robotic manipulators that are required to repeat a given task, chemical batch processes or, more generally, the class of tracking systems.

Since the original work Arimoto, Kawamura, and Miyazaki (1984), the general area of ILC has been the subject of intense research effort. Initial sources for the literature here are the survey papers Bristow, Tharayil, and Alleyne (2006) and Ahn, Chen, and Moore (2007). In ILC, a major objective is to achieve convergence of the trial-to-trial error. It is, however, possible that enforcing fast convergence could lead to unsatisfactory performance along the trial, and here this problem is addressed by first showing that ILC schemes can be designed for a class of discrete linear systems by extending techniques developed for linear repetitive processes (Rogers, Galkowski, & Owens, 2007). This allows us to use the strong concept of stability along the pass (or trial) for these processes, in an ILC setting, as a possible means of dealing with poor/unacceptable transients in the dynamics produced along the trials. The results developed give control law design algorithms that can be implemented via LMIs, and results from their experimental implementation on a gantry robot executing a pick and place operation are also given. Finally, it is shown how the analysis can be extended to robust control where the uncertainty is associated with along the trial dynamics, and again supporting experimental results are given.

Throughout this paper $M \succ 0$ (respectively, $\prec 0$) denotes a real symmetric positive (respectively, negative) definite matrix. Also the identity and null matrices of the required dimensions are denoted by I and 0, respectively.

2. Background

Consider a case when the plant to be controlled can be modeled as a differential linear time-invariant system with state-space model defined by $\{A_c, B_c, C_c\}$. In an ILC setting this is written as

$$\dot{x}_k(t) = A_c x_k(t) + B_c u_k(t), \quad 0 \le t \le \alpha$$

$$y_k(t) = C_c x_k(t) \tag{1}$$

where on trial k, $x_k(t) \in \mathbb{R}^n$ is the state vector, $y_k(t) \in \mathbb{R}^m$ is the output vector, $u_k(t) \in \mathbb{R}^r$ is the vector of control inputs, and $\alpha < \infty$ is the trial length. If the signal to be tracked is denoted by $y_{ref}(t)$ then $e_k(t) = y_{ref}(t) - y_k(t)$ is the error on trial k, and the most basic requirement is to force the error to converge as $k \to \infty$. In particular, the objective of constructing a sequence of input functions such that the performance is gradually improving with each successive trial can be refined to a convergence condition on the input and error

$$\lim_{k\to\infty} \|e_k\| = 0, \quad \lim_{k\to\infty} \|u_k - u_\infty\| = 0 \tag{2}$$

☆ This work has been partially supported by the Ministry of Science and Higher Education in Poland under the Project N N514 293235.

* Corresponding author.
E-mail addresses: L.Hladowski@issi.uz.zgora.pl (L. Hladowski), K.Galkowski@issi.uz.zgora.pl (K. Galkowski), zc@ecs.soton.ac.uk (Z. Cai), etar@ecs.soton.ac.uk (E. Rogers), cf@ecs.soton.ac.uk (C.T. Freeman), pll@ecs.soton.ac.uk (P.L. Lewin).

0967-0661/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2009.12.003

where $\|\cdot\|$ is a signal norm in a suitably chosen function space with a norm-based topology.

It is, however, possible that trial-to-trial convergence will occur but produce dynamics along the trials which is far from satisfactory for many practical applications. Consider, for example, a gantry robot executing the following set of operations: (i) collect an object from a location and place it on a moving conveyor, (ii) return to the original location and collect the next one and place it on the conveyor, and (iii) repeat (i) and (ii) for the next one and so on. Then if the object has an open top and is filled with liquid, and/or is fragile in nature, unwanted vibrations during the transfer time could have very detrimental effects. Hence in such cases there is also a need to control along the trial.

One approach to the analysis of ILC schemes with the potential to address the dynamics in both directions, i.e. trial-to-trial and along the trial, is to use a 2D systems setting. There has already been work in this direction using the well known Roesser (1975) and Fornasini and Marchesini (1978) state-space models. For example, in Kurek and Zaremba (1993) it was shown how trial-to-trial error convergence of linear ILC schemes in the discrete domain could be examined as a stability problem in terms of a Roesser state-space model interpretation of the dynamics. To-date, however, relatively little attention has been directed towards control law design in a 2D systems setting for both error-to-error convergence and along the trial dynamics, with the exception of Shi, Gao, and Wu (2005) for the robust control case. In this last publication, the ILC control law is first applied to the process and then the uncertainty structure is assumed for the resulting model. This is somewhat artificial and here the more natural case when the uncertainty is associated with the uncontrolled process model is considered and therefore the need to deal with uncertainty associated with products of matrices is avoided. Also the robust control laws here are experimentally verified and the level of control action monitored.

There has been a considerable volume of work on robust ILC for discrete processes, see, for example, van de Wijdeven and Bosgra (2008) in the case of trial-to-trial uncertainty. Most of the work reported in this area uses the concept of lifting for ILC systems described by discrete linear time-invariant state-space models, see, for example, Ahn, Moore, and Chen (2007) for background, which uses stacked state, output and input vectors to write the ILC dynamics in a standard linear systems setting. This involves the products of state-space model matrices, which is very difficult to handle in uncertainty characterization.

Given that the trial length is finite by definition, it follows that ILC fits naturally into the class of so-called repetitive processes (Rogers et al., 2007). The unique characteristic of a repetitive, or multipass, process is a series of sweeps, termed passes, through a set of dynamics defined over a fixed finite duration known as the pass length. On each pass an output, termed the pass profile, is produced which acts as a forcing function on, and hence contributes to, the dynamics of the next pass profile. This, in turn, leads to the unique control problem that the output sequence of pass profiles generated can contain oscillations that increase in amplitude in the pass-to-pass direction.

To introduce a formal definition, let $\alpha<\infty$ denote the pass length (assumed constant). Then in a repetitive process the pass profile $y_k(t)$, $0\le t\le\alpha$, generated on pass k acts as a forcing function on, and hence contributes to, the dynamics of the next pass profile $y_{k+1}(t)$, $0\le t\le\alpha$, $k\ge 0$.

Attempts to control these processes using standard systems theory and algorithms fail (except in a few very restrictive special cases) precisely because such an approach ignores their inherent 2D systems structure, i.e. information propagation occurs from pass-to-pass (k direction) and along a given pass (t direction) and also the initial conditions are reset before the start of each new pass. To remove these deficiencies, a rigorous stability theory has been developed (Rogers et al., 2007) based on an abstract model of the dynamics in a Banach space setting that includes a very large class of processes with linear dynamics and a constant pass length as special cases, including those considered in this paper.

To introduce the required background on the abstract model, let E_α be a Banach space and W_α a linear subspace of W_α. Suppose also that $y_k\in E_\alpha$ is the pass profile on pass k. Then the process dynamics are described by linear recursion relations of the form

$$y_{k+1}=L_\alpha y_k+b_{k+1},\quad k\ge 0 \tag{3}$$

where L_α is a bounded linear operator mapping E_α into itself and b_{k+1} represents known initial conditions, disturbance and control input effects on pass $k+1$.

Consider now discrete linear repetitive processes described by the following state-space model over $p=0,1,\ldots,\alpha-1,\ k\ge 0$,

$$x_{k+1}(p+1)=Ax_{k+1}(p)+Bu_{k+1}(p)+B_0y_k(p)$$

$$y_{k+1}(p)=Cx_{k+1}(p)+Du_{k+1}(p)+D_0y_k(p) \tag{4}$$

where on pass k, $x_k(p)\in\mathbb{R}^n$ is the state vector, $y_k(p)\in\mathbb{R}^m$ is the pass profile vector, $u_k(p)\in\mathbb{R}^r$ is the control input vector, α is the finite pass length. To complete the process description, it is necessary to specify the initial, or boundary, conditions, i.e. the state initial vector on each pass and the initial pass profile. Here these are taken to be zero. To write this process in abstract model terms, regard the pass profile on pass k as the ordered set $y_k=\{y_k(0),y_k(1),\ldots,y_k(\alpha)\}$ regarded as a point in the product space $E_\alpha=\{R^m\times R^m\times\cdots\times R^m\}$ with norm $\|y_k\|=\max_{0\le p\le\alpha}\|y_k(p)\|'$ (where $\|\cdot\|'$ denotes the norm in R^m). Then L_α is the convolution operator for the linear discrete-time system defined by the state-space quadruple (state, input, output and direct feedthrough matrix, respectively) $\{A,B_0,C,D_0\}$.

In the next section, it is shown how a repetitive process setting can be used to analyze ILC schemes and, in particular, how the stability theory of these processes can be employed to develop algorithms for control law design for trial-to-trial error convergence and along the trial performance.

3. ILC analysis and control law design

From this point onwards, the discrete domain is considered and hence it is assumed that the process dynamics have been sampled by the zero-order hold method at a uniform rate T_s seconds to produce a discrete state-space model with matrices $\{A,B,C\}$. Also rewrite the state equation of the process model in the form

$$x_k(p)=Ax_k(p-1)+Bu_k(p-1) \tag{5}$$

and introduce

$$\eta_{k+1}(p+1)=x_{k+1}(p)-x_k(p) \tag{6}$$

$$\Delta u_{k+1}(p)=u_{k+1}(p)-u_k(p) \tag{7}$$

Then

$$\eta_{k+1}(p+1)=A\eta_{k+1}(p)+B\Delta u_{k+1}(p-1) \tag{8}$$

Consider also a control law of the form

$$\Delta u_{k+1}(p)=K_1\eta_{k+1}(p+1)+K_2e_k(p+1) \tag{9}$$

and hence

$$\eta_{k+1}(p+1)=(A+BK_1)\eta_{k+1}(p)+BK_2e_k(p) \tag{10}$$

Also $e_{k+1}(p)-e_k(p)=y_k(p)-y_{k+1}(p)$ and hence

$$e_{k+1}(p)-e_k(p)=CA(x_k(p-1)-x_{k+1}(p-1))+CB(u_k(p-1)-u_{k+1}(p-1)) \tag{11}$$