ELSEVIER

Annual Reviews in Control 29 (2005) 57–70

Annual Reviews in Control

www.elsevier.com/locate/arcontrol

Iterative learning control — An optimization paradigm

D.H. Owens *, J. Hätönen

Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

Received 3 December 2004; received in revised form 24 January 2005; accepted 27 January 2005
Available online 21 April 2005

Abstract

The area if iterative learning control (ILC) has emerged from robotics to form a new and exciting challenge for control theorists and practitioners. There is great potential for applications to systems with a naturally repetitive action where the transfer of data from repetition (trial or iteration) can lead to substantial improvements in tracking performance. Although many of the challenges met in control systems design are familiar, there are several serious issues arising from the "2D" structure of ILC and a number of new problems requiring new ways of thinking and design. This paper introduces some of these issues from the point of view of the research group at Sheffield University and concentrates on linear systems and the potential for the use of optimization methods to achieve effective control.

Keywords: Iterative learning control; Optimization; Robust control

1. Introduction

Iterative learning control is a relatively new technique for improving tracking response in systems that repeat a given task over and over again (each repetition sometimes being called a *pass* or *trial*). As in standard tracking problems, as a starting point a plant model

$$x(t+1) = \Phi x(t) + \Gamma u(t), \quad x(0) = x_0,$$
$$y(t) = Cx(t) + Du(t) \tag{1}$$

is given where Φ, Γ, C and D are matrices of appropriate dimensions, and $u(\cdot)$ is the input variable, $x(\cdot)$ the state variable, x_0 the initial condition for $x(\cdot)$, and $y(\cdot)$ the output variable. For notational simplicity, it is assumed that $D = 0$. Furthermore, a reference signal $r(t)$ is given over a finite time-interval $t \in [0, T]$, and the objective is to find a control input $u(t)$ so that the corresponding output $y(t)$ tracks $r(t)$ as accurately as possible. The assumption that differentiates ILC from the standard tracking problem is the following: when the system (1) has reached the final time point $t = T$, the state x of the system is reset to the *same* initial condition x_0, and, after resetting, the system is again required to track the *same* reference signal $r(\cdot)$. At first sight this assumption might look artificial, but several important industrial applications do indeed fit into the ILC framework. Reported applications of ILC include robotics (Norrlöf, 2002; Zilouchian, 1994), chemical batch processing (Lee, Bang, Yi, & Yoon, 1996a), and servo systems (Lee & Lee, 1996), to name but a few. For a detailed discussion on ILC applied to industrial problems, see Longman (2000).

In the past (and presumably still in most of the cases nowadays, at least in industry), a control system for any system satisfying the ILC assumptions is set up once and then remains unchanged. In other words the control action $u(t)$ is determined from some form of fixed feedback control, which results in a control action $u(t) = u_f(t)$. The problem, however, is that (disturbances and noise to one side) the feedback controller will produce the same input function $u_f(\cdot)$ during every repetition, and hence if the corresponding output function $y_f(t) = [Gu_f](t)$ is not equal to $r(t)$ for each $t \in [0, T]$, the resulting *non-zero* tracking error $e_f(t)$ is repeated exactly and without change in each repetition. Consequently, in Arimoto, Kawamura, and Miyazaki (1984), possibly motivated by human learning, it was suggested that information from the repetitions $1, 2, \ldots, k-1$ could be used to construct a new, improved input function u_k, where k is the repetition number. He also

* Corresponding author.
E-mail addresses: d.h.owens@sheffield.ac.uk (D.H. Owens), j.hatonen@sheffield.ac.uk (J. Hätönen).

1367-5788/$ – see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.arcontrol.2005.01.003

demonstrated the feasibility that this can be done in a way that ensures that the tracking error will ultimately go to zero as the number of repetitions increases. In summary, ILC has the property that experience from previous repetitions or iterations can be used to ensure that the ILC system will gradually learn the control action that will result in perfect tracking. An extremely appealing possibility!

Remark 1. Note that in the ILC community it is now widely accepted that (Uchiyama, 1978) is the first publication to introduce the ILC concept. However, because this publication is written in Japanese, non-Japanese researchers were not aware of this publication when the ILC research initially started in USA and Western Europe. Therefore, the publication (Arimoto et al., 1984) was referenced as being the starting point for ILC research. However, it seems that there are earlier publications than Uchiyama's on topics related to ILC. For example, in Cryer, Nawrocki, and Lund (1976) the authors (who worked for General Motors Truck and Bus) proposed the following discrete-time 'iterative control' (as they call it) in the frequency domain

$$u_{k+1}(e^{j\omega T_s}) = u_k(e^{j\omega T_s}) + \beta G^{-1}(e^{j\omega T_s})e_k(e^{j\omega T_s}) \tag{2}$$

where $\beta \in \mathbb{R}, 0 < \beta < 1$ is a design parameter, and $G(e^{j\omega T_s})$ is the discrete-time Fourier transform of the impulse response $g(t)$ of the plant in question. This is clearly an ILC algorithm and the authors apply this algorithm in the context of laboratory road simulation. Another even earlier reference to ILC concepts seems to be the USA Patent 3,555,252 — Learning Control of Actuators in Control Systems from 1971, see Chen and Moore (2000) for details.

Example 1. In order to clarify the difference between familiar feedback control and ILC, consider a dynamical system

$$(p^2 + 5p + 6)y(t) = (p + 1)u(t) \tag{3}$$

where $p := \mathrm{d}/\mathrm{d}t$ and $t \in [0, 6]$. This system is sampled at intervals of $T_s = 0.1$ s using zero-order hold resulting in a discrete-time plant model (Φ, Γ, C). The system is required to track a reference signal $r(t) = \sin(2\pi t/6)$ and the system is controlled with a PID-controller

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\,\mathrm{d}\tau + K_D \dot{e}(t) \tag{4}$$

where $K_p = 15$, $K_I = 8$ and $K_D = 0$, which is also sampled using zero-order hold with the same sampling time. Fig. 1 shows the output $y(t)$, implying that the system is only capable of tracking the reference signal to a moderate degree of accuracy. Note that this same tracking result is obtained during each repetition, because the PID-controller has fixed parameters. Fig. 2, on the other hand, shows the l_2-norm (Euclidean norm) of the tracking error $e_k(t) := r(t) - y_k(t)$

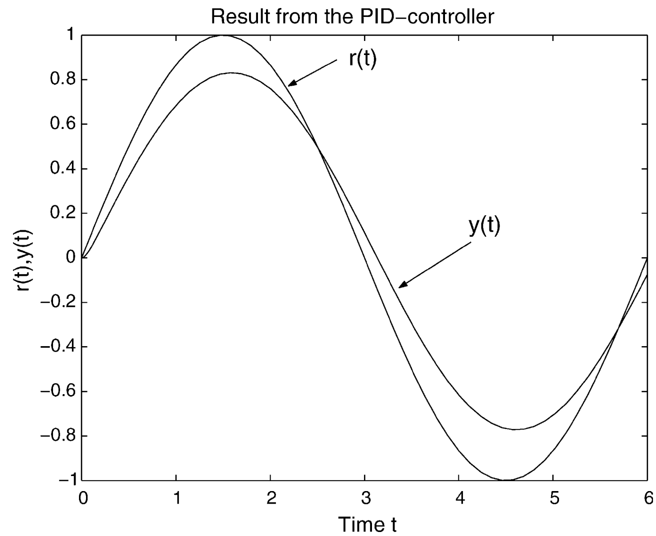

Fig. 1. The response for the PID-controller (4).

as a function of the iteration index (number or round) k with the ILC algorithm

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t + 0.1) \tag{5}$$

where $\gamma = 9$. Note that at first sight this algorithm seems to be non-causal, because $u_{k+1}(t)$ is a function of $e_k(t + 0.1)$. However, on iteration $k + 1$, $e_k(t)$ is *past* data and is available for the whole range $t \in [0, T]$. In general, it is possible (and typically necessary) to make $u_{k+1}(t)$ a function of $e_k(s)$ for $s \geq t$. Based on Fig. 2 it seems that the tracking error does indeed tend to zero as $k \to \infty$, but the convergence is not monotonic in the sense that errors do increase in early repetitions but ultimately reduce to zero. In the following section it will be shown that this algorithm converges to zero tracking error for an arbitrary time-invariant linear plant (assuming that $C\Gamma \neq 0$) if the learning gain γ satisfies the inequality $|1 - \gamma C\Gamma| < 1$.

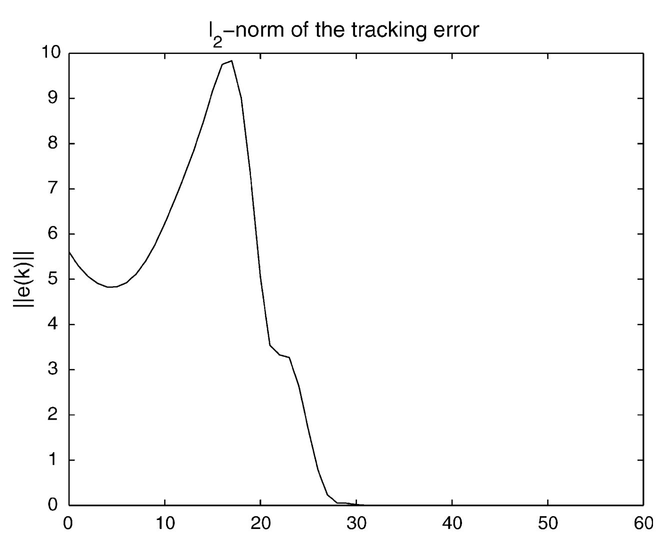

Fig. 2. Convergence behaviour of the Arimoto-algorithm.