Control Engineering Practice 57 (2016) 39–49

Contents lists available at ScienceDirect

Control Engineering Practice

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

A new method for analysis and design of iterative learning control algorithms in the time-domain

Norbert Zsiga*, Stijn van Dooren, Philipp Elbert, Christopher H. Onder

Institute for Dynamic Systems and Control, ETH Zurich, Zurich 8092, Switzerland

ARTICLE INFO

Article history:
Received 26 November 2015
Received in revised form
23 August 2016
Accepted 23 August 2016

Keywords:
Iterative learning control
Convergence rate
Internal combustion engine
Filter
Fourier series

ABSTRACT

In this paper, a novel analysis method for iterative learning control (ILC) algorithms is presented. Even though expressed in the lifted system representation and hence in the time-domain, the convergence rate as a function of the frequency content of the error signal can be determined. Subsequently, based on the analysis method, a novel ILC algorithm (F-ILC) is proposed. The convergence rate at specific frequencies can be set directly in the design process, which allows simple tuning and a priori known convergence rates. Using the F-ILC design, it is shown how to predict the required number of iterations until convergence is achieved, depending on the reference trajectory and information on the system repeatability. Numerical examples are given and experimental results obtained on an internal combustion engine test bench are shown for validation.

1. Introduction

Learning from experience is important for humans to improve skills and abilities for a variety of tasks in everyday life. Observations of errors are used to improve one's performance for the next time. In technical systems, for repetitive tasks, iterative learning control (ILC) algorithms constitute a learning process. The deviation of the system output from a specified reference trajectory is calculated after every iteration (trial). The error is used to calculate a modified input signal for the next iteration to improve the system performance. While non-learning controllers keep repeating the same error, ILC reduces the error in each iteration as the controller learns which input signal leads to the smallest possible error.

Iterative Learning Control methods are used for a variety of control processes in research and industry such as chemical batch reactors (Lee, Lee, & Kim, 2000; Liu, Gao, & Wang, 2010), high-performance maneuvers of quadrocopters (Hehn & D'Andrea, 2013; Schoellig, Mueller, & D'Andrea, 2012), 3D printing (Bolder, Oomen, & Steinbuch, in press), suppression of residual vibrations (van de Wijdeven & Bosgra, 2008), and industrial robots (Mahamood & Pedro, 2011; Hakvoort, Aarts, van Dijk, & Jonker, 2008). For further areas of application, the reader is referred to the recent survey papers by Ahn, Chen, and Moore (2007), Bristow, Tharayil, and Alleyne (2006), Wang, Gao, and Doyle (2009), and Xu (2011). There exist several methods to design ILC algorithms that are based in the time-domain or in the frequency domain. In addition to that, a 2D system representation for design and analysis is also possible as in Hladowski et al. (2008), Hladowski et al. (2010), and Bolder et al., in press.

Convergence of the algorithms has been addressed in the majority of publications since the first publications of ILC in 1984 (Arimoto, Kawamura, & Miyazaki, 1984). The convergence rate is defined as the ratio of the errors of two successive iterations under a given norm. In Tang, Cai, and Huang (2000), Zuo, Zhu, and Cai (2009), Oh and Lee (2014), Owens and Hätönen (2005), Bristow et al. (2006), Arif, Ishihara, and Inooka (2003), Moore, Chen, and Ahn (2006) the correlation between the learning gain(s), i.e. tuning parameter(s) of the ILC algorithms, and the convergence rate is discussed, but only qualitatively and/or an upper bound is given.

Amann, Owens, and Rogers (1996) derive a gradient type algorithm that is based on optimization. The authors state that the convergence rate is initially much faster than in later iterations, but they offer no explanation for this observation. In Ghosh and Paden (2002), it is qualitatively stated that the pseudo-inverse based learning approach leads to a faster convergence of low-frequency components and a slow convergence of high-frequency components of the error.

If the learning process is designed in the frequency-domain, the information that can be extracted from the algorithm is different from that obtained if the learning is designed in the time-domain. The authors Kim, Zou, and Su (2008) use a frequency dependent learning gain (iterative coefficient) to quantify the frequency range

* Corresponding author.
E-mail address: nzsiga@ethz.ch (N. Zsiga).

http://dx.doi.org/10.1016/j.conengprac.2016.08.014
0967-0661/© 2016 Published by Elsevier Ltd.

of convergence but do not comment on the convergence rate. In Goh (1994) a frequency-domain method is presented with which the convergence rate of one single frequency can be set directly. The convergence rate for all other frequency components in the error signal is also determined by this choice. Freeman, Lewin, and Rogers (2007) analyze the convergence rate of frequency-domain iterative learning algorithms as a function of the frequency. Longman (2000) analyzes the growth or decay of single frequency components of the error signal, and this analysis is done in the frequency domain.

Typically, the articles mentioned discuss the fact that the convergence rate can be altered by increasing or decreasing some tuning parameter(s). Often, an upper bound for the convergence rate is given, but a quantitative description of the convergence rate is usually lacking. For frequency-domain ILC algorithms, the method mentioned in Freeman et al. (2007) provides an insight into the frequency-dependent convergence behavior of an algorithm. In Dinh, Freeman, and Lewin (2014), Owens (2016), the convergence properties of gradient and norm-optimal ILC are discussed. Dinh et al. (2014) provide bounds for the error reduction after j iterations, whereas in Owens (2016), a connection is described between approximate eigenvectors and the convergence rate of single frequency components in the error signal. However, neither of these two papers mention whether the analyses are also applicable to lifted system matrices of more general ILC design procedures.

This paper contains two novelties. First, an analysis method is introduced that offers an insight into the frequency-dependent convergence rate that is achieved using various ILC algorithms in the time-domain. The convergence rate depends on the system under investigation, the tuning parameters of the ILC algorithm, as well as on the given reference trajectory. The method allows the convergence rate to be estimated as a function of the frequency content of the reference trajectory. It is applicable to any ILC algorithm in the lifted system representation. Non-causal filters, expressed in matrix notation, are presented that are based on the discrete Fourier transform. The frequency characteristics of the filters can be chosen freely. This type of filter is subsequently used for the design of a new ILC algorithm (F-ILC), which is the second novelty. With the F-ILC algorithm a monotonic convergence is guaranteed, and the frequency-dependent convergence rate can be specified in the design process. This is an advantage in view of the fact that no further analysis is necessary to determine the behavior of the system concerning stability and convergence properties.

The paper is structured as follows: After an overview of the system description, the conditions for stability and monotonic convergence of ILC algorithms are presented. The popular Q-ILC algorithm is then illustrated with a numerical example. Based on the simulation results, the method is presented to determine the frequency-dependent convergence rate in the time-domain representation. Subsequently, a so-called frequency matrix is introduced and its link to the discrete Fourier transform is explained. The properties of the matrix are used for the design of the F-ILC algorithm. Numerical and experimental results for this algorithm are shown. The experiments are carried out on an internal combustion engine test bench. The conclusions are presented in the last section.

2. System description

A discrete, linear, and time-invariant (LTI) SISO system $\Sigma(z)$ can be described by

$$x(k+1) = \tilde{A}x(k) + \tilde{B}u(k) \tag{1}$$

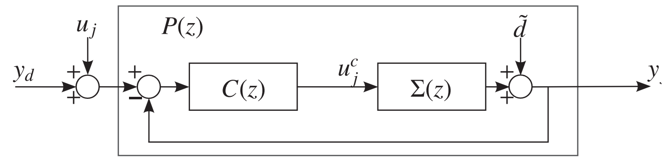

Fig. 1. Structure of the control system.

$$y(k) = \tilde{C}x(k) \tag{2}$$

where k is the discrete-time index. Fig. 1 shows the structure of the control system. The controller is denoted by $C(z)$ while the complementary sensitivity is denoted by $P(z)$.

Considering finite discrete signals of length N, the following time history vectors are defined

$$y_j = [y_j(1) \quad y_j(2) \quad \cdots \quad y_j(N)]^T \tag{3}$$

$$y_d = [y_d(1) \quad y_d(2) \quad \cdots \quad y_d(N)]^T \tag{4}$$

$$u_j = [u_j(0) \quad u_j(1) \quad \cdots \quad u_j(N-1)]^T \tag{5}$$

$$\tilde{d} = [\tilde{d}(0) \quad \tilde{d}(1) \quad \cdots \quad \tilde{d}(N-1)]^T \tag{6}$$

where j is the iteration index (Bristow et al., 2006; Phan, Longman, Panomruttanarug, & Lee, 2013). The signals y_j and y_d are shifted by one time step to account for the one-step delay of the plant. The iteration-invariant reference is denoted by y_d, the current iteration ILC control input by u_j, and the output by y_j. In the 0th iteration $u_0 = [0 \quad 0 \quad 0 \quad \cdots \quad 0]^T$ is used. The controller $C(z)$ generates the signal u_j^c within the iteration. However, this signal is not relevant for ILC as only the input and output of $P(z)$ must be known. The signal $\tilde{d}$ captures iteration-invariant disturbances and initial conditions of the system (Phan et al., 2013). Using a matrix P defined as

$$P = \begin{bmatrix} CB & & & & \\ CAB & CB & & & \\ CA^2B & CAB & \ddots & & \\ \vdots & \vdots & \ddots & CB & \\ CA^{N-1}B & \cdots & CA^2B & CAB & CB \end{bmatrix}, \tag{7}$$

the system dynamics can be captured by the static equation

$$y_j = P(y_d + u_j) + d, \tag{8}$$

with the vector $d = S(z)\tilde{d}$ and where the variable $S(z)$ is the sensitivity of the control loop. The Markov parameters of the system can be found in the first column of P (Hespanha, 2009). This system representation is often referred to as lifted system representation (Bristow et al., 2006; Janssens et al., 2012; Schoellig et al., 2012). The error vector of the current iteration is defined as

$$e_j = y_d - y_j. \tag{9}$$

In the remainder of the paper the lifted system representation is used. In this notation, a general first-order ILC update law can be written as

$$u_j = Q\left(u_{j-1} + Le_{j-1}\right), \tag{10}$$

where L is the learning gain matrix and Q is the lifted-system matrix of a filter. The goal of using ILC algorithms is to design a learning gain matrix L such that the algorithm is asymptotically stable and the error converges monotonically.