Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-7 (2016) 342–346

Design of a Data-Driven Controller for a Spiral Heat Exchanger

Shin Wakitani * **Mingcong Deng** ** **Toru Yamamoto** ***

* *Tokyo University of Agriculture and Technology, Tokyo, 184-8588 Japan (e-mail: wakitani@cc.tuat.ac.jp).*
** *Tokyo University of Agriculture and Technology, Tokyo, 184-8588 Japan (e-mail: deng@cc.tuat.ac.jp)*
*** *Hiroshima University, Hiroshima, 739-8527, Japan (e-mail: yama@cc.tuat.ac.jp)*

Abstract: A data-driven proportional-integral-derivative (DD-PID) controller has been proposed as an effective controller for nonlinear systems. The DD-PID controller can tune the PID parameters adaptively at each equilibrium point. In order to train the PID parameters in a database, an offline learning algorithm based on a fictitious reference iterative tuning (FRIT) method was established. This method can compute the PID parameters by using a set of operating data. However, the FRIT method is a control parameter tuning method that is only based on the minimization of the system output in its criterion; therefore, the criterion is insufficient for systems in which the stability of a closed-loop system is important such as chemical process systems because sometimes the sensitivity of an obtained controller becomes high. In order to solve this problem, an extended FRIT (E-FRIT) method that penalizes the input variation in its criterion has been proposed. In this method, the PID parameters that are taken into stability can be calculated. The effectiveness of the proposed method is evaluated by an experimental result of a spiral heat exchanger.

Keywords: Process control, PID control, self-tuning control, data-driven control, extended fictitious reference iterative tuning (E-FRIT)

1. INTRODUCTION

Proportional-integral-derivative (PID) controllers (Ziegler and Nichols (1942); Chien et al. (1952); Vilanova and Visioli (2012)) have been mainly used in many process systems because of its simple structure. The control performance of a PID controller is strongly affected by the combination of three parameters: the proportional gain, the integral gain and the derivative gain. If a control system has nonlinearity, a fixed PID controller may not maintain the desired control performance when the equilibrium point of the system output is changed by altering the set points. A data-driven PID (DD-PID) controller that uses a database for tuning PID parameters has been proposed by Yamamoto et al. (2009) as an effective controller for such nonlinear systems.

The DD-PID controller is one of a just-in-time controller (Stenman et al. (1996); Zheng and Kimura (2001); Nakpong and Yamamoto (2012)). The DD-PID controller adaptively tunes its PID parameters at each equilibrium point by using trained PID parameters that are stored in a database. Learning algorithms of the DD-PID controllers are divided into two methods: one is an online learning method that trains PID parameters while under control, and the other is an offline learning method that trains the parameters in advance by using obtained experimental data. There is a method based on a fictitious reference iterative tuning (FRIT) method (Kaneko et al. (2005)), which is one of the offline learning methods. This method is known as the DD-FRIT method (see Wakitani et al. (2013)). According to the method, the time required for learning a database is significantly reduced because this method can update the database using one-shot experimental I/O data.

In the FRIT method, a fictitious reference signal including obtained experimental I/O data and adjustable control parameters is generated firstly. After that, a desired controller is realized by adjusting the control parameters in order to minimize the error of the system output in its criterion. The FRIT method does not require any mathematical models of the controlled object to tune the control parameters. However, control parameters tuned by the FRIT method may become high gain because the FRIT method is only based on the minimization of the control output. In order to solve this problem, an extended FRIT (E-FRIT) method that has a penalty to compensate for the differential input is proposed (see Masuda et al. (2010), Kano et al. (2011)). The E-FRIT method can help suppress the sensitivity of a controller by setting a penalty factor appropriately.

In this paper, a new offline learning method of the DD-PID controller based on the E-FRIT method is proposed. In this method, the PID gain updating rule based on the E-FRIT is derived. A more stable DD-PID controller can be obtained compared to the previous learning method based on the FRIT method. In this study, the proposed

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.320

DD-PID controller is applied to a spiral heat exchanger and its effectiveness is evaluated.

2. DESIGN OF DATA-DRIVEN CONTROLLER

2.1 System Description

It is assumed that the controlled object is described as the following equation.

$$y(t) = f(\boldsymbol{\phi}(t-1)), \tag{1}$$

where, $y(t)$ is the system output and $f(\cdot)$ indicates a nonlinear function whose output is determined by a historical data vector $\boldsymbol{\phi}(t-1)$ (See Yamamoto et al. (2009)). The historical data $\boldsymbol{\phi}(t-1)$ denotes as follows.

$$\begin{aligned}\boldsymbol{\phi}(t-1) := [&y(t-1), \ldots, y(t-n_y),\\ &u(t-1), \ldots, u(t-n_u)].\end{aligned} \tag{2}$$

In (2), $u(t)$ is the system input, n_y and n_u are orders of $y(t)$ and $u(t)$, respectively.

2.2 PID Control Law

When a PID controller is applied to process systems, sometimes derivative kick depending on set value change causes problems for stability of a process. In order to avoid the derivative kick, this paper introduces the following PID control law. This control law is known as I-PD control law.

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t), \tag{3}$$

where, $e(t)$ is the control error between the set value $r(t)$ and the system output $y(t)$, and is defined as

$$e(t) := r(t) - y(t). \tag{4}$$

In (3), $K_P(t)$, $K_I(t)$ and $K_D(t)$ express the proportional gain, the integral gain and the derivative gain, respectively. Moreover, Δ denotes the differencing operator given by $\Delta := 1 - z^{-1}$, and z^{-1} is the backward operator which implies $z^{-1}y(t) = y(t-1)$. In the DD-PID method, these PID gains at each step t are adaptively tuned using a database.

2.3 Data-driven PID controller

This section explains the working principle of the DD-PID controller. In the DD-PID controller, an initial database has to be created because the controller requires a database for its actions. Thus, if a database does not exist, an initial database is created by the following procedure.

[STEP 1] Generate Initial Database

Initial operating data $r_0(t), u_0(t), y_0(t)$ are obtained by using an I-PD controller with fixed PID gains. Datasets at each step are generated by obtained operating data, and are sequentially stored in the database. The dataset is defined by the following equation.

$$\boldsymbol{\Phi}(j) = [\bar{\boldsymbol{\phi}}(t_j), \boldsymbol{\theta}_{PID}(t_j)], \quad j = 1, 2, \ldots, N. \tag{5}$$

Where, N indicates the total number of datasets. $\bar{\boldsymbol{\phi}}(t_j)$ and $\boldsymbol{\theta}_{PID}(t_j)$ are expresses as follows.

$$\begin{aligned}\bar{\boldsymbol{\phi}}(t_j) := [&r_0(t_j+1), r_0(t_j),\\ &y_0(t_j), \ldots, y_0(t_j - n_y + 1),\\ &u_0(t_j - 1), \ldots, u_0(t_j - n_u + 1)]\end{aligned} \tag{6}$$

$$\boldsymbol{\theta}_{PID}(t_j) = [K_P(t_j), K_I(t_j), K_D(t_j)]. \tag{7}$$

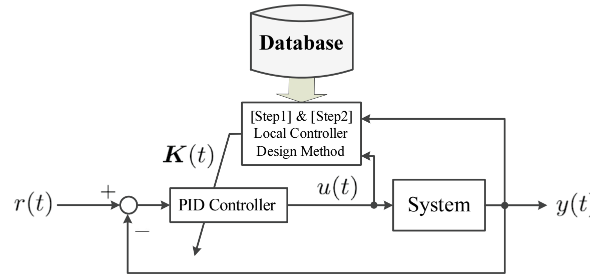

Fig. 1. Block diagram of the data-driven proportional-integral-derivative (DD-PID) control system.

PID gains at each step t are calculated by the following [STEP 2] and [STEP 3].

[STEP 2] Calculate Distance and Select Neighbor Data

The distance between the query (which is the information vector that indicates current system state) $\bar{\boldsymbol{\phi}}(t)$ and an information vector $\bar{\boldsymbol{\phi}}(t_j)$ in the database is calculated by the following $\mathcal{L}_1$ norm with some weights.

$$d(\bar{\boldsymbol{\phi}}(t), \bar{\boldsymbol{\phi}}(t_j)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \bar{\phi}_l(t_j)}{\max \bar{\phi}_l(m) - \min \bar{\phi}_l(m)} \right|, \tag{8}$$

$$j = 1, \ldots, N.$$

In (8), $\bar{\phi}_l(t)$ expresses the l-th element in the j-th dataset, and $\bar{\phi}_l(t)$ expresses the l-th element in the query at t. Moreover, $\max \bar{\phi}_l(m)$ and $\min \bar{\phi}_l(m)$ indicate the maximum value and minimum value of the l-th element of all datasets in the database. In this method, the datasets in the database are sorted in ascending order of their distance, and k-pieces of datasets with the smallest distances among them are chosen as neighbor datasets. Where, k is set by the user at will.

[STEP 3] Compute PID gains

From the selected k-pieces of neighbor datasets, a suitable set of PID gains at t steps are computed by the following equation.

$$\boldsymbol{K}(t) = \sum_{i=1}^{k} w_i \boldsymbol{K(i)}, \qquad \sum_{i=1}^{k} w_i = 1, \tag{9}$$

where

$$w_i = \frac{\exp(-d_i)}{\displaystyle\sum_{i=1}^{k} \exp(-d_i)}. \tag{10}$$

The block diagram of the DD-PID controller is shown in Fig. 1. By executing [STEP 2] and [STEP 3] every time, the PID gains are adaptively tuned if the PID gains in the database are suitably tuned in advance. However, if a result obtained by a fixed PID controller is applied to create a database, then all PID gains included in the initial information vectors may be equal. Expressed numerically, that is

$$\boldsymbol{\theta}_{PID}(1) = \boldsymbol{\theta}_{PID}(2) = \cdots = \boldsymbol{\theta}_{PID}(N). \tag{11}$$

In this case, the PID gains in the database have to be tuned in an offline manner or online manner. The online learning method requires many experiments to get optimal PID