Design of A Parameter Update Method of the Database-Driven PID Controller Considering H_∞ Norm of the System

Takumi Ueda
Kyushu institute of technology, 2-4 Hibikino, Wakamatsu, Kitakyushu, Fukuoka, 808-0135, Japan

Kazuo Ishii
Kyushu institute of technology, 2-4 Hibikino, Wakamatsu, Kitakyushu, Fukuoka, 808-0135, Japan

Abstract

PID controllers are used in many applications, and they are considered as typical examples of fixed-structure controllers. For each specific system, the PID parameters of the controller must be adjusted appropriately. Therefore, database-driven control, which automatically adjusts the parameters of the PID controller using a large amount of experimental data, has attracted attention. Database-driven control is one of the methods that does not require model information. Therefore, in the Database-driven PID (DD-PID), the Jacobian of the system used for the steepest descent method is unknown. In this paper, we present an optimization method using the Gradient descent method based on the local model of the system. And we proposed a method that can suppress the increase of H_∞ norm more than the conventional DD-PID method.

Keywords: Database-Driven PID, H_∞ norm, Nonlinear system

1. Introduction

PID controllers are widely used in industry [1] and are one of the typical examples of controllers with a fixed structure. To adjust the parameters appropriately, it is effective to identify a mathematical model of the system, and many methods have been reported so far [2]. However, the systems in general use have various nonlinearities, making it difficult to construct a detailed model, and the desired control performance may not be achieved with a conventional fixed PID controller. Therefore, database-driven control, in which PID controller parameters are automatically adjusted according to system characteristics using a large amount of experimental data, has been actively studied. [3],[4],[5],[6]

The DD-PID method, one of the database-driven control methods, constructs a local model based on data accumulated in a database and updates the PID gains using a nonlinear optimization method. The conventional method uses the just-in-time (JIT) method to construct the local model, which calculates the local model by extracting data from the data stored in the database that are similar to the query created using the current operation data as the nearest neighbor data. Appropriate selection of neighboring data and design of the database and query is important to construct a local model using the JIT method, and they also affect the control results.

In this paper, we examine a database design method for the JIT method, which identifies the local model of the system using the Recursive Least Squares (RLS) method and reflects the H_∞ norm of the calculated local model in the database. Furthermore, the Jacobian of the local model identified by RLS is reflected in the gradient method to optimize the PID gain update. Finally, the proposed method is verified by numerical simulations.

2. Proposed method

The DD-PID method proposed in this section updates the PID gains by the following steps.

Step.1 Setting of the reference model and experiment using initial PID gains

The DD-PID method cannot update parameters when there is no data in the database. Therefore, an initial database is constructed using a stable controller. The database used in this paper is defined by equation (1), where $j = 1, \cdots, N_d(0)$ and $N_d(0)$ is the number of databases in the initial state. In addition, $\|M(j)\|_\infty$ is the H_∞ norm of the local model calculated using the RLS method described below, ϕ_d is the information vector, and θ_d is the gain vector of the controller.

$$\boldsymbol{\Phi}(j) \coloneqq [\boldsymbol{\phi}(j), \|M(j)\|_\infty, \boldsymbol{\theta}_d(j)] \quad (1)$$

Define ϕ_d and θ_d by equations (2) and (3). where N_y and N_u are the orders of y and u.

$$\boldsymbol{\phi}_d := [y(j), \cdots, y(j - N_y), u(j), \cdots, u(j - N_u)] \quad (2)$$

$$\boldsymbol{\theta}_d(j) \coloneqq [K_P(j), K_I(j), K_D(j)] \quad (3)$$

The reference model for the DD-PID method is defined by equations (4) and (5). where μ is a parameter related to attenuation, ρ is a parameter related to rising time, and T_s is the sampling time.

$$y_m(t) = \frac{P(1)z^{-1}}{P(z)} = \frac{(1 + p_1 + p_2)z^{-1}}{1 + p_1 z^{-1} + p_2 z^{-2}} r(t) \quad (4)$$

$$\begin{cases} p_1 = -2e^{-(2\mu)^{-1}\rho} \cos\left((2\mu)^{-1}\rho\sqrt{4\mu - 1}\right) \\ p_2 = e^{-\rho\mu^{-1}} \\ \mu \coloneqq 0.25(1 - \delta) + 0.51\delta \\ \rho \coloneqq \sigma^{-1} T_s \end{cases} \quad (5)$$

Step. 2 Getting queries and neighborhood data

Obtain the query q while operating the control target. q is defined by equations (6) and (7).

$$\boldsymbol{q}(t) \coloneqq [\boldsymbol{\phi}(t), \|M(t)\|_\infty] \quad (6)$$

$$\boldsymbol{\phi}(t) := [y(t), \cdots, y\left(t - N_y\right), u(t), \cdots, u(t - N_u)] \quad (7)$$

Where $\|M(t)\|_\infty$ is the H_∞ norm of the local system calculated by the RLS method. the system to be estimated by RLS is defined by equation (8).

$$H_{RLS}(z) = \frac{b_1(t)z^{-1} + b_2(t)z^{-2}}{1 + a_1(t)z^{-1} + a_2(t)z^{-2}} \quad (8)$$

Using the retrieved query, the proximity distance of the data to the information vector in the database is calculated, and neighboring data are selected. Although several algorithms have been proposed for selecting nearby data from a large amount of data, this paper uses Kinoshita's[7] method for obtaining nearby data based on similarity. The similarity between the information vector and the query is given by equations (9) and (10).

$$\boldsymbol{S}\left(\phi_d(j), q(t)\right) = \prod_{i=1}^{N_y+N_u+1} \frac{1}{\sqrt{2\pi h_i^2}} e^{-\frac{\alpha^2}{2}} \quad (9)$$

$$\alpha = h_i^{-1}\left(q_i(t) - \phi_{d,i}(j)\right) \quad (10)$$

where h_i is the bandwidth of the probability density function, $q_i(t)$ is the i-th element of the query at time t, and $\phi_{d,i}(j)$ is the i-th element of the j-th information vector in the database. The selection of neighboring data is based on similarity according to equation (11). The designer sets $T_{th} (0 \le T_{th} \le 1)$. For example, if $T_{th} = 0.95$, data that match 95% of the queries are selected from the database.

$$\boldsymbol{S}\left(\boldsymbol{\phi}_d(j), \boldsymbol{q}(t)\right) \ge T_{th} \prod_{i=1}^{N_y+N_u+1} \frac{1}{\sqrt{2\pi h_i^2}} \quad (11)$$

The results of neighborhood data selection are also used to update the database.

Step. 3 Local model calculation by JIT method

Using the neighborhood data selected in Step.2, construct a local model using Equations (12) and (13) to obtain the PID gain $\tilde{\theta}$.

$$\tilde{\boldsymbol{\theta}}(t) = \sum_{i=1}^{n_k} \boldsymbol{\omega}_i \boldsymbol{\theta}(i), \quad \sum_{i=1}^{n_k} \boldsymbol{\omega}_i = 1 \quad (12)$$

$$\boldsymbol{\omega_i} = \frac{\boldsymbol{S}\left(\boldsymbol{\phi}_d(j), \boldsymbol{q}(t)\right)}{\sum_{j=1}^{n_k} \boldsymbol{S}\left(\boldsymbol{\phi}_d(j), \boldsymbol{q}(t)\right)} \quad (13)$$

The gain $\tilde{\theta}$ calculated by the JIT method may be inappropriate. Therefore, the gain $\tilde{\theta}$ calculated in step 3 is corrected using the gradient method. The gain correction by the gradient method is given by Equation (14). As shown in equation (14), the gain correction by the DD-PID method is delayed by one step.

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(t) - \boldsymbol{\eta} \frac{\partial J(t+1)}{\partial \tilde{\boldsymbol{\theta}}(t)} \quad (14)$$

Where η is the learning coefficient for the gradient method, J is the evaluation function, and ε is the difference between the system output and the reference model output, defined by equations (15) - (17), respectively.

$$\boldsymbol{\eta} \coloneqq [\eta_P, \eta_I, \eta_D] \quad (15)$$

$$J(t) \coloneqq 0.5\varepsilon(t)^2 \quad (16)$$

$$\varepsilon(t) = y(t) - y_m(t) \quad (17)$$

From these relationships, the partial derivative in equation (14) can be expanded using the differential chain rule as in equation (18).

$$\frac{\partial J(t+1)}{\partial\, \tilde{\boldsymbol{\theta}}(t)} = \frac{\partial J(t+1)}{\partial \varepsilon(t+1)} \frac{\partial \varepsilon(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial\, \tilde{\boldsymbol{\theta}}(t)} \quad (18)$$

The Jacobian of the system is given by equation (19) from equation (8).

$$\frac{\partial y(t+1)}{\partial u(t)} = b_1(t) \quad (19)$$

Step. 4 Update the database

Since the DD-PID method updates gains based on data stored in the database, the amount of data stored in the database should be kept to a minimum. Therefore, the query and PID gains are added to the database only when no neighboring data are selected. This reduces the amount of similar data and thus is expected to reduce the amount of computation.

Step. 5 Evaluation and Convergence Decision

Update the controller parameters by repeating Step.2 through Step.5 until the evaluation function shown in equation (20) becomes sufficiently small or is repeated a specified number of times.

$$\boldsymbol{J}(epoch) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(y_0(i) - y_r(i)\right)^2} \qquad (20)$$

3. *Simulation*

3.1. *System parameter settings*

In this paper, the effectiveness of the proposed method is verified by numerical simulations using Hammerstein's nonlinear model shown in Equations (21) and (22). [8]

$$\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad + 1.2x(t-1) - 0.1x(t-2) + \xi(t) \end{aligned} \qquad (21)$$

$$x(t) = \begin{cases} 1.0u - 1.5u^2 + 1.0u^3 (t < 70) \\ 1.0u - 1.0u^2 + 1.0u^3 (t \geq 70) \end{cases} \qquad (22)$$

Target values were set as in equation (23).

$$r(t) = \begin{cases} 0.5(\quad 0 < t \leq \ 50) \\ \ \ 1(\ 50 < t \leq 100) \\ \ \ 2(100 < t \leq 150) \\ 1.5(150 < t \leq 200) \end{cases} \qquad (23)$$

The reference model is designed as in equation (24), where $\delta = 0$ and $\sigma = 10$.

$$P(z) = 1 - 0.271z^{-1} + 0.0183z^{-2} \qquad (24)$$

Other parameters are shown in Table 1.

Table 1. Setting parameters for DD-PID

Order of the information vector	$N_y = 5$ $N_u = 5$
The constant value for RLS	$\alpha = 1000$ $\boldsymbol{P}_0 = \alpha \boldsymbol{I}$ $\boldsymbol{\zeta}_0 = \boldsymbol{O}$
Similarity threshold	$T_h = 0.9$
Number of forgetting coefficient	$\rho = 0.99$
Number of initial databases	$N(0) = 6$

3.2. *Simulation result*

The initial database is constructed using the initial PID gain shown in equation (25).

$$K_P = 0.486, K_I = 0.227, K_D = 0.112 \qquad (25)$$

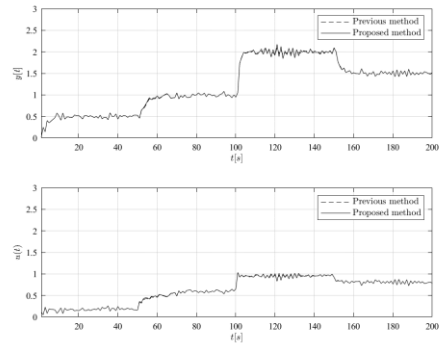

Fig.1 comparison of output between the proposed and previous method

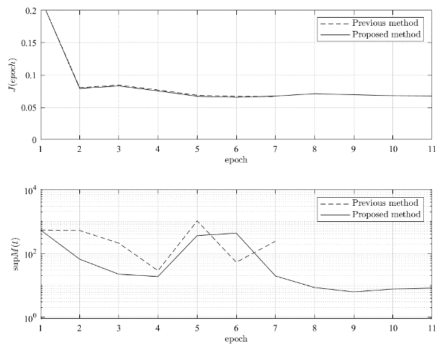

Fig.2 Result of b_1 by RLS

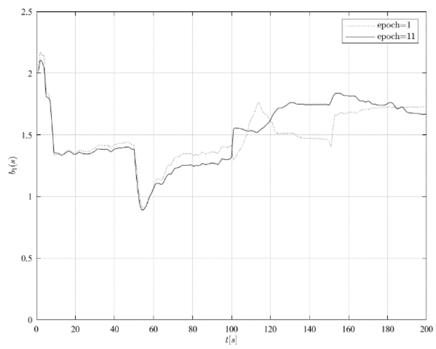

Fig.3 Result of the evaluation function and H_∞ norm

Fig. 1 shows the control results of the conventional and proposed methods, Fig. 2 shows the results of the evaluation function and H_∞ norm, and Fig. 3 shows b_1 estimated by RLS. Fig. 2 shows that the value of the evaluation function of the conventional method is 0.067, while that of the proposed method is 0.068. On the other hand, the H_∞ norm is 240.35 for the conventional method and 8.165 for the proposed method. Fig. 3 shows that the Jacobian b_1 of the system is always positive, but varies widely.

4. *Condition*

In this paper, we propose a method to adjust the learning coefficient of the gradient method appropriately based on the H_∞ norm obtained from the model identified by RLS, and a method to improve the stability of the system by adding information on the local H_∞ norm to the database. The effectiveness of the proposed method was confirmed through numerical simulations by adapting it to the Hammerstein-type nonlinear model discussed in a previous study.

Future work is to investigate a method for directly specifying the H_∞ norm and a method for real-time updating of PID gains when database-driven control is used.

References

1. N. Suda, “PID seigyo (PID control), Asakura syoten, 1992, (In Japanese).
2. S. Adachi, ”Sisutemu seigyo no kiso(Basic system identification)”, Tokyo Denki University Press, 2009,(In Japanese).
3. S. Wakitani, K. Nishida, M. Nakamoto, T. Yamamoto, “Design of a Data-Driven PID Controller using Operating Data,” IFAC Proceedings Volumes, Vol.46, No.11, pp-587-592, 2013.
4. K. Takao, T. Yamamoto, T. Hinamoto, ″A Design of Memory-Based PID Controllers,″ Transactions of the Society of Instrument and Control Engineers, Vol.40, No.9, pp.898-905, 2004,(In Japanese).
5. R. Kurozumi, T. Yamamoto, “A Design of CMAC Based Intelligent PID Controllers”, IEEJ Transactions on Electronics, Information and Systems, Vol.125,No.4, pp.607-615, 2005,(In Japanese).
6. S. Wakitani, Y. Ohnishi, T. Yamamoto, ”Design of a CMAC-Based PID Controller Using FRIT for Nonlinear Systems,” Transactions of the Society of Instrument and Control Engineers,Vol.48,No.12,pp.847-853, 2012,(In Japanese).
7. T. Kinoshita,T. Yamamoto, ”Design of a Database-Driven Control System based on the Similarity with Reducing Calculation Cost”, IEEJ Transactions on Electronics, Information and Systems,Vol.140, No.3, pp.312-319, 2020,(In Japanese).
8. T. Yamamoto, O. Kaneko, Y. Ohnisi, S. Wakitani, T. Kinoshita , H. Kugemoto, ”Data shikougata PID seigyo(Data-Driven PID control)”,Morikita publishing CO.LTD, 2020,(In Japanese).

Authors Introduction

Mr. Takumi Ueda

He received his M.E., in Engineering from Kyusyu Institute of Technology, Japan, in 2017. He is a third-year student in the doctoral program at the Kyushu Institute of Technology. His research interest includes non-linear control by PID using database.

Dr. Kazuo Ishii

He is a Professor in the Kyushu Institute of Technology, where he has been since 1996. He received his Ph.D. degree in engineering from University of Tokyo, Tokyo, Japan, in 1996. His research interests span both ship marine engineering and Intelligent Mechanics. He holds five patents derived from his research.