

Study on a Sub-databases-driven (S-DD) Controller using k-means Clustering

Shin Wakitani* Hiroki Nakanishi* Toru Yamamoto*

* Hiroshima University, 1-4-1 Kagamiyama, Higashihiroshima,
Hiroshima 739-8527 Japan
(e-mail: wakitani/yama@hiroshima-u.ac.jp).

Abstract: A database-driven PID (DD-PID) control method is one of the effective control methods for nonlinear systems. In the conventional DD-PID control method, there is a problem that the calculation cost and required memory for creating an optimal database are large. For the above problem, this paper proposes a method to implement the DD-PID controller with small-sized sub-databases. In the proposed method, one database that includes past I/O data and PID gains are created, and the database is updated in an offline manner. Moreover, sub-databases are constructed by clustering the created database using the k-means clustering method. The number of clusters for k-means clustering is determined automatically based on kernel functions. The effectiveness of the proposed method is presented by numerical examples.

Keywords: PID control, Learning control, Database management systems, Learning algorithms, Nonlinear control.

1. INTRODUCTION

A PID controller has been widely used in real industrial systems because of its simple structure. However, since many industrial systems have nonlinearity, it is difficult to maintain desired control performance by setpoint changes and/or environment changes. A database-driven PID (DD-PID) controller design method (Yamamoto et al. (2009); Wakitani et al. (2019)) is one of the practical solutions for such systems. The DD-PID controller design method follows the idea of Just-in-Time (JIT) modeling (Stenman et al. (1996)). The controller generates a local PID controller every step based on a query, a vector which indicates the current system state by the current I/O data, and dataset that include past I/O data and adopted PID gains together stored into a database. Specifically, a distance between a query and an dataset is calculated to measure the similarity of both data, and dataset with smaller distances are extracted from a database as neighbor data. Eventually, PID gains of the local controller are calculated by PID gains included in the extracted neighbor data.

To obtain a good control performance of a DD-PID controller, following TWO important points must be considered.

- (1) A database that includes optimal control parameters at each equilibrium point is must be constructed to maintain a good local controller. The first question is how to create such an optimal database by a few experiments.
- (2) While controlling, the control parameters of a local controller must be calculated within a determined sample time. The second question is how to reduce the calculation time of creating a local controller.

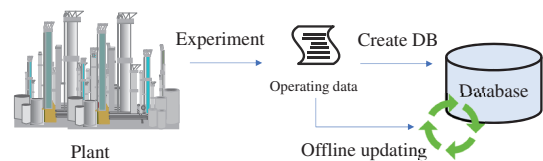


Fig. 1. Database tuning method using 1-shot closed-loop control data.

In this paper, the above two questions are solved by the following approach. For the first question, a method that adopts a data-driven approach represented by the virtual reference feedback tuning (VRFT) (M.C.Campi (2002)) and the fictitious reference iterative tuning (FRIT) (Kaneko (2013)) to updating PID gains stored into a constructed database is proposed. According to the method, the PID gains into a database can be updated by using 1-shot closed-loop control data shown in Fig. 1.

For the second question, a sub-database-driven (S-DD) PID control approach that is one of the expanded algorithms of the DD-PID controller is proposed. In the proposed method, one database is divided into some small sub-databases by the k-means clustering method (MacQueen (1967)). Where, in the k-means clustering method, it usually becomes a problem that how to determine the number of divided clusters in advance. In this paper, the method that can automatically determine the number of divided clusters based on a kernel function (Gramacki (2018)) is also proposed. The proposed S-DD-PID controller switches the sub-databases in an online manner as shown in Fig. 2. The calculation cost for extracting and calculating PID gains can be reduced compared with the conventional DD-PID controller with one database.

This paper is organized as follows. In section 2 shows how to design a sub-databases for a S-DD PID controller. Es-

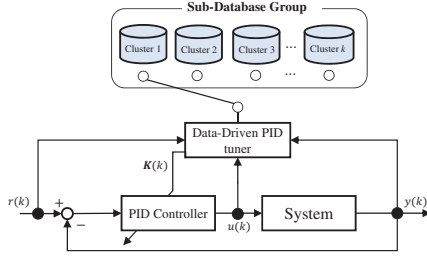


Fig. 2. Proposed sub-database-driven PID control system. pecially, the sub-databases construction methods based on the k-means clustering method and a kernel function are particularly explained. In section 3, the offline updating method of PID gains based on the FRIT is explained. According to this scheme, optimal PID gains in a database can be obtained by 1-shot experimental data set without any system models. In section 4, the effectiveness of the proposed method is presented by numerical examples.

2. SUB-DATABASES-DRIVEN PID CONTROLLER DESIGN

2.1 System Description

It is assumed that the nonlinear system is described by the following equation.

$$y(t) = f(\phi(t-1)), \quad (1)$$

where t and $y(t)$ denote the step time and the system output. $f(\cdot)$ is a nonlinear function whose output is determined by a historical data vector $\phi(t-1)$. The historical data $\phi(t-1)$ are denoted as follows.

$$\phi(t-1) := [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]. \quad (2)$$

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

2.2 PID Control Law

When a PID controller is applied to process systems. This paper introduces the following velocity-type PID control law that can avoid the derivative kick. This control law is known as the I-PD control law.

$$\Delta u(t) = K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \quad (3)$$

where $e(t)$ is the control error which is defined as follows:

$$e(t) := r(t) - y(t). \quad (4)$$

In (3), $K_P(t)$, $K_I(t)$ and $K_D(t)$ indicate 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 $z^{-1}y(t) = y(t-1)$. $r(t)$ indicates the reference signal. In the DD-PID method, these PID gains at each step are determined by JIT approach using a database. The detail explanation of the calculation of PID gains based on the JIT approach is given in section 2.5.

2.3 Creating Initial Database

Initial operating data r_0, u_0 and y_0 , that is, the reference signal, the control input and the system output, are obtained by using an I-PD controller with fixed PID gains.

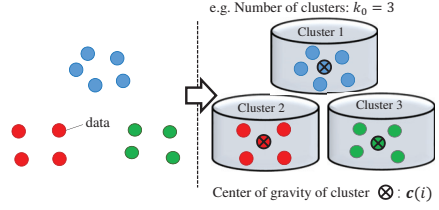


Fig. 3. Schematic of k-means clustering.

Datasets at each step are generated by using the obtained operating data and are sequentially stored in the database. Dataset Φ is defined by the following equation.

$$\Phi(j) = [\bar{\phi}_j^T, \theta_j^T], \quad j = 1, 2, \dots, N, \quad (5)$$

where j and N denote the index of the dataset and the total number of datasets, respectively. The dataset has two sections: $\bar{\phi}(t)$ is called the information vector that expresses the state of the controlled object at t . $\theta(t)$ expresses PID gains vector which is adopted to the local PID controller at t . These vectors are given as follows:

$$\bar{\phi}_j := [r_0(t+1), r_0(t), y_0(t), \dots, y_0(t-n_y+1), u_0(t-1), \dots, u_0(t-n_u+1)]^T \quad (6)$$

$$\theta_j = [K_P(t), K_I(t), K_D(t)]^T. \quad (7)$$

Note that, if the result obtained by using a fixed PID controller is applied to create a database, then all PID gains included in the initial datasets may be equal. This can be expressed numerically as follows:

$$\theta_1 = \theta_2 = \dots = \theta_N. \quad (8)$$

To obtain good control performance by DD PID controllers, it is important to optimally tune the above PID gains in advance. The updating algorithm of the PID gains will be explained in section 3.

2.4 Sub-databases Design based on k-means Clustering and Kernel Function

The k-means clustering is one of the non-layered clustering methods using unsupervised learning. The schematic of the k-means clustering is shown in Fig. 3. The data group is automatically classified into k_0 clusters by the following algorithm.

- (I) Number of clusters k_0 and initial values of centroids $c(i)$ ($i = 1, 2, \dots, k_0$) are set. Where, the dimension of the $c(i)$ is same as $\bar{\phi}_j$ in (6). Moreover, k_0 is set as large value to unify clusters. The detail of the clusters unification will be explained later.
- (II) Calculate distance between each $\bar{\phi}_j$ and each centroid of clusters $c(i)$, and allocate dataset $\Phi(j)$ to the cluster with the shortest distance cluster centroid c .
- (III) Calculate the value of the centroid including allocated data.
- (IV) Repeat (I) to (III) until the value of all centroids are not changed.

The clustered data groups are regarded as sub-databases.

The conventional k-means clustering must be determined the number of clusters k_0 in advance. This paper proposes a simple and automatically determining method of the number of clusters based on a kernel function.

After generating k_0 numbers of clusters, the similarity of each cluster is calculated by the following kernel function:

$$f(\mathbf{c}(m), \mathbf{c}(i)) = \prod_{l=1}^{n_y+n_u+1} \frac{1}{\sqrt{2\pi}H_l} \exp\left\{-\frac{(\mathbf{c}_l(m) - \mathbf{c}_l(i))^2}{2H_l^2}\right\} \quad (9)$$

Where, $\mathbf{c}_l(m)$ ($m = 1, 2, \dots, k_0$) represents the l -th element of the centroid of the m -th cluster. H_l is a bandwidth representing the smoothness of the distribution function, and is determined by the plug-in method. Specifically, the H_l can be calculated as follows:

$$H_l = \frac{1.06\sigma_i}{N^{\frac{1}{5}}}. \quad (10)$$

σ_i is the standard distribution defined as follows:

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{n=1}^N (\mathbf{c}(m) - \mu_i)^2}. \quad (11)$$

Note that the similarity expressed by (9) becomes maximized when $\mathbf{c}(i) = \mathbf{c}(m)$, that is $\prod_{l=1}^{n_y+n_u+1} \sqrt{2\pi}H_l$. Select the cluster whose values of the centroid are bigger than a certain threshold based on the following formula:

$$f(\mathbf{c}(m), \mathbf{c}(i)) \geq \frac{T_k}{\prod_{l=1}^{n_y+n_u+1} \sqrt{2\pi}H_l} \quad (12)$$

$$(m, i = 1, 2, \dots, k)$$

Where, T_k ($0 \leq T_k \leq 1$) is the threshold. Apply (9) and (12) up to $m = 1, 2, \dots, k$, and the centroid $\mathbf{c}(i)$ of clusters that are more than the threshold and data contained in each cluster are integrated into one cluster. Define the remaining number of integrated clusters as k . Each cluster centroid value after integration is the average of the integrated cluster centroids. Users can set thresholds based on similarity, and the preset number of clusters k_0 is integrated into k .

As described above, the sub-databases clustered by k -means are integrated based on the similarity.

2.5 PID Gains Tuning based on Sub-databases

The S-DD-PID controller shown in Fig. 2 can be designed by using the constructed sub-databases. While controlling, the S-DD-PID controller generates a local PID controller based on JIT approach. The local PID controller can be generated by following steps.

[STEP 1] Select Sub-databases

Firstly, the query $\bar{\phi}(t)$ is obtained at t step. After that, the distance between query and each centroid is calculated by following equation.

$$d_i(\bar{\phi}(t), \mathbf{c}(i)) = \sum_{l=1}^{n_y+n_u+1} \left| \frac{\bar{\phi}_l(t) - \mathbf{c}_l(i)}{\max_m \mathbf{c}_l(i) - \min_m \mathbf{c}_l(i)} \right|, \quad (13)$$

$$i = 1, \dots, k.$$

In (13), $\bar{\phi}_l(t_j)$ expresses the l -th element in the j -th dataset, and $\bar{\phi}_l(t)$ expresses the l -th element in the query. Moreover, $\max_m \mathbf{c}_l(m)$ and $\min_m \mathbf{c}_l(m)$ indicate the maximum and minimum values of the l -th element of all the centroids. The centroid that has minimum distance with the

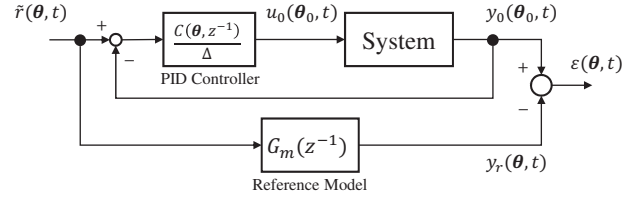


Fig. 4. Block diagram of fictitious reference iterative tuning.

query is chosen, and all of the attached datasets with the centroid are extract as neighbor data.

[STEP 2] Compute PID Gains

A PID gains of a local PID controller at t are computed by using the following equation using selected neighbor data.

$$\theta^*(t) = \sum_{j=1}^{N_i} w_j \theta_j, \quad \sum_{j=1}^{N_i} w_j = 1, \quad (14)$$

where, N_i is number of datasets stored in i -th sub-database (centroid). Moreover, the weights w_j are calculated as follows:

$$w_j = \frac{\exp(-d_j)}{\sum_{j=1}^{N_i} \exp(-d_j)}. \quad (15)$$

[STEP 3] Return to STEP 1

3. OFFLINE UPDATING ALGORITHM OF SUB-DATABASES

This section explains an offline PID gains updating algorithm based on the fictitious reference iterative tuning (FRIT). FRIT method is firstly presented, and an algorithm of PID gains that stored in the database constructed in section 2 is explained.

3.1 Fictitious Reference Iterative Tuning (FRIT)

FRIT method calculates the control parameters of a linear controller directly using a set of closed-loop data obtained by a stable controller. Although FRIT can calculate any control parameters of a linear controller, this paper only explain a tuning principal of a PID controller.

Let's consider the closed loop system shown in Fig. 4. Firstly, it supposes that $u_0(\theta, t)$ and $y_0(\theta, t)$ are obtained by a fixed PID controller $C_0(\theta_0, z^{-1})/\Delta$ with an initial control parameters vector $\theta_0 = [K_{P0}, K_{I0}, K_{D0}]^T$. Where, $C_0(\theta_0, z^{-1})$ is given as the following polynomial,

$$C_0(\theta_0, z^{-1}) = K_{I0} + K_{P0}z^{-1} + K_{D0}z^{-2}. \quad (16)$$

The relationships among r_0 , y_0 , u_0 , and $C_0(\theta_0, z^{-1})/\Delta$ can be described as follows.

$$u_0(\theta, t) = \frac{C_0(\theta_0, z^{-1})}{\Delta} \{r_0(\theta, t) - y_0(\theta, t)\}. \quad (17)$$

By rewriting (17), the fictitious reference signal $\tilde{r}(\theta, t)$ is generated by tuned PID gains ($\theta = [K_P, K_I, K_D]^T$) as follows.

$$\tilde{r}(\theta, t) = C^{-1}(\theta, z^{-1})\Delta u_0(\theta, t) + y_0(\theta, t). \quad (18)$$

FRIT solves the following optimization problem and derives the optimal control parameters ($\theta^* = [K_P^*, K_I^*, K_D^*]^T$).

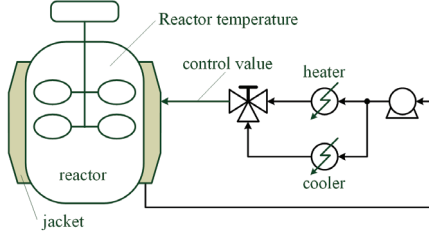


Fig. 6. Schematic of Polystyrene Reactor.

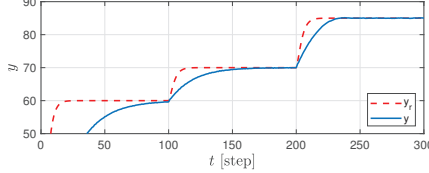


Fig. 7. Control result obtained by fixed PID controller.

STEP 4: Calculating $\tilde{r}(t)$ and $y_r(t)$ by using (18) and (25).

STEP 5: Calculating correction terms by (28) and $\theta^T(t)$ by using (22).

STEP 6: Updating PID gains in the sub-database by using (31).

STEP 7: Repeating from Step1 to Step 6 until value of (23) at each step becomes sufficiently small.

4. NUMERICAL EXAMPLE

4.1 Controlled object

This simulation deals with a polystyrene reactor model shown in Fig. 6. The control objective is to control the reactor temperature by manipulating the jacket temperature. A mathematical relationship between the reactor temperature and jacket temperature is given as follows:

$$y(t) = 0.804y(t-1) + 5.739 \times 10^{15} \cdot \exp\{-E_a/R(y(t-1) + 273)\} + 0.148u(t-1) + \xi(t). \quad (32)$$

where $\xi(t)$ denotes a Gaussian white noise with zero mean and variance of 0.001^2 . Moreover, the sampling interval is set to $T_s = 1$ s. The reference signals were set as follows:

$$r(t) = \begin{cases} 60 & (0 \leq t < 100) \\ 70 & (100 \leq t < 200) \\ 85 & (200 \leq t < 300) \end{cases}. \quad (33)$$

In the simulations, the characteristic polynomial $P(z^{-1})$ of the reference model in 25 is expressed as

$$P(z^{-1}) = 1 - 1.34z^{-1} + 0.449z^{-2}. \quad (34)$$

4.2 Fixed PID control

A PID controller with fixed PID gains was applied. The fixed PID gains were calculated based on the Chien, Hrones, and Reswick (CHR) method (Vilanova and Visioli (2012)) as follows:

$$K_P = 9.0, \quad K_I = 0.5 \quad K_D = 1.0. \quad (35)$$

The control result obtained is shown in Fig. 7. Note that, in Fig. 7, y_r represents the reference signal outputted from

Table 1. Design parameters.

Variable	Value	Description
n_y	3	Order of output variables
n_u	5	Order of input variables
k_0	20	Initial number of clusters
k	6	Unified number of clusters
T_k	0.01	Threshold amount for unifying clusters

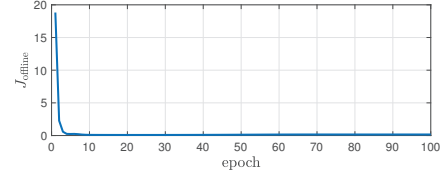


Fig. 8. Trajectory of learning error in offline learning.

the reference model. This result shows that the transient properties of the system output at different reference signals are not the same. It implies the difficulty of obtaining good control performance at each equilibrium point using the fixed PID controller because of the nonlinearity of the system. All the obtained closed-loop data and fixed PID gains were transformed according to the format of the datasets in (5), and an initial database was created from these datasets.

4.3 Sub-databases design and offline updating

Sub-databases are designed by following a design procedure explained in section 2.4. Where the design parameters of the proposed method is summarized in Table 1. From the table, the number of clusters are reduced from $k_0 = 20$ to $k = 6$. It implies that the proposed unifying scheme explained in section 2.4 worked properly.

After creating the sub-databases, PID gains in the sub-databases are updated by the proposed offline updating scheme explained in section 3. Where, the learning coefficients are set as follows:

$$\eta_P = 10, \quad \eta_I = 0.025, \quad \eta_D = 0.01. \quad (36)$$

After setting the parameters, offline updating was executed by utilizing the closed-loop data that was used for generating the sub-databases. As stated above, this method requires a number of iterations to complete the updating. Thus, in order to confirm whether updating has been completed, the following error function was introduced and monitored:

$$J(\text{epoch}) = \frac{1}{N} \sum_{k=1}^N \{y_0(k) - y_r(k)\}^2 \quad (37)$$

The error behaviors of offline updating are shown in Fig. 8. The figure shows that updating progressed significantly at approximately 5 epochs, and convergence was almost achieved at 10 epochs. In this simulation, the sub-databases updated 100 epochs, that is satisfactorily updated shown in Fig. 8, are adopted.

4.4 Proposed S-DD PID control

The control result by using the proposed S-DD controller is shown in Fig. 9. Moreover, the trajectories of selected

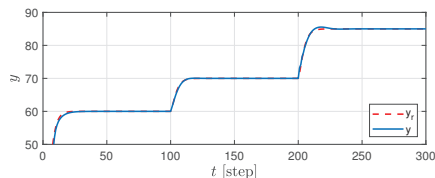


Fig. 9. Control result obtained by proposed method.

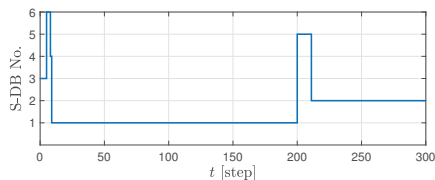


Fig. 10. Trajectories of selected sub-databases no..

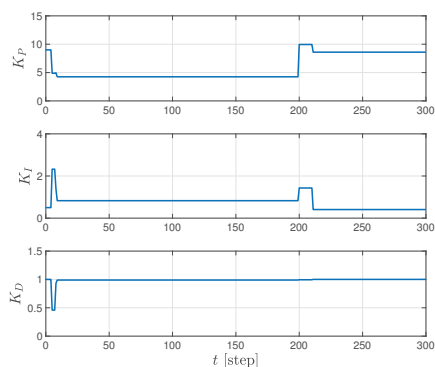


Fig. 11. Trajectories of PID gains corresponding to Fig. 9.

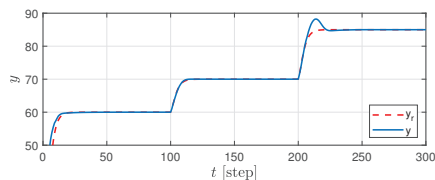


Fig. 12. Control result obtained by a conventional DD-PID controller with one database.

sub-databases no. and PID gains are shown in Fig. 10 and Fig. 11, respectively.

These results show that good control result can be obtained by selecting sub-databases. Especially, the sub-databases are switched in the transient state, and PID gains are tuned greatly.

4.5 Comparison with conventional method

Finally, a comparison between using only one database and using sub-databases are shown below. The control result only using one database are shown in Fig. 12 and Fig. 13. Where, the learning coefficients in offline learning are the same as (36). These result shows that the PID gains, especially the P gain, is changing frequently. However, the larger overshoot is occurred around where $y = 85$ comparison with the result of the proposed method.

The average calculation time in each step was measured by 'tic' and 'toc' functions provided by MATLAB 2019a.

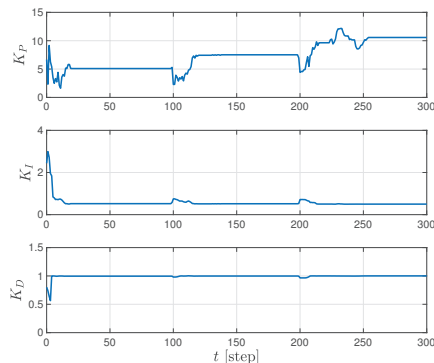


Fig. 13. Trajectories of PID gains corresponding to Fig. 12.

Using only one database : 2.2184×10^{-4} s

Using Sub – databases : 1.1844×10^{-4} s

The above comparisons show that the proposed method can reduce calculation time about a half. With these comparisons, the effectiveness of the proposed method is verified.

ACKNOWLEDGEMENTS

This work was supported by ' Hiroshima Manufacturing Digital Innovation Creation Program ' with the Grant from Cabinet Office, Government of Japan and Hiroshima Prefecture.

REFERENCES

- Gramacki, A. (2018). Nonparametric kernel density estimation and its computational aspects. *Springer International Publishing*.
- Kaneko, O. (2013). Data-driven cotnroller tuning: Frit approach. *Proc. of 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, 326–336.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, 1, 281–297.
- M.C.Campi (2002). Virtual reference feedback tuning (vrft) : A direct method for the design of feedback controllers. *Automatica*, 38(8), 1337–1346.
- Stenman, A., Gustafsson, F., and Ljung, L. (1996). Just in time models for dynamical systems. *35th IEEE Conference on Decision and Control*, 1115–1120.
- Vilanova, R. and Visioli, A. (2012). *PID control in the Third Millennium : lessons learned and new approaches*. Springer.
- Wakitani, S., Yamamoto, T., and Bhushan, G. (2019). Design and application of a database-driven pid controller with data-driven updating algorithm. *Industrial & Engineering Chemistry Research*, 58(26), 11419–11429.
- Yamamoto, T., Takao, K., and Yamada, T. (2009). Design of a data-driven pid controller. *IEEE Trans. Control Systems Technology*, 17(1), 29–39.