Title	Direct Data-Driven Tuning of Look-Up Tables for Feedback Control Systems
Author(s)	Yahagi, Shuichi; Kajiwara, Itsuro
Citation	IEEE Control Systems Letters, 6, 2966-2971 https://doi.org/10.1109/LCSYS.2022.3181343
Issue Date	2022-06-08
Doc URL	http://hdl.handle.net/2115/86496
Rights	© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Type	article (author version)
File Information	20220604_Paper3_fnl_wo_margin.pdf

Direct data-driven tuning of look-up tables for feedback control systems

Shuichi Yahagi and Itsuro Kajiwara

***Abstract*—In industry, a feedback controller with a look-up table (LUT) is often used for nonlinear systems. Although this structure is easy to understand, tuning the LUT parameters is time-consuming due to the huge number of parameters. This paper presents a direct data-driven design method for a gain-scheduled feedback controller with a LUT to directly tune the LUT parameters from single-experiment data without a system model. Specifically, conventional virtual reference feedback tuning (VRFT), which is a data-driven method, is extended and the L_2 norm for adjacent LUT parameters is added to the VRFT cost function to avoid overlearning. The optimized parameters are analytically obtained by a generalized ridge regression. A simulation of a nonlinear system demonstrates that the proposed method can directly obtain the LUT parameters without knowledge of the controlled object's characteristics.**

***Index Terms*— Data-driven control, VRFT, Gain-scheduled control, Look-up table, Tuning**

I. Introduction

MORE than 90% of closed-loop controls in industrial systems use classical proportional–integral–derivative (PID) control [1]. Classical PID control is easy to understand and provides suitable performances for strongly linear systems. However, sufficient control performance is difficult to achieve with fixed gains for nonlinear systems. Nonlinear control theory and model-based control can be applied, but the hurdles are high due to the limited controller performance, complex theory, and large computational load. Additionally, model-based control may not be effective because the intricacy of industrial systems prevents accurate modeling.

Against this background, gain-scheduled PID control using look-up tables (LUTs) is often used in industry [2, 3]. Gain-scheduled control achieves the desired control performance by adjusting the controller parameters according to the plant's state. Although the use of LUTs is an intuitive approach, numerous parameters must be tuned to achieve the desired performance [2]. Fixed PID control requires tuning of only three parameters, whereas many parameters must be tuned for gain-scheduled control using LUTs. Hence, parameter tuning is time consuming. This is a crucial problem in industry.

The Ziegler-Nichols (ZN), the Chien-Hrones-Reswick (CHR), and model-based methods may be applicable to controller tuning. However, the ZN and the CHR methods are empirical and model-based methods depend on the identified mathematical model. Recently, direct data-driven controller tuning methods have been investigated as they do not need a system model to be controlled. Virtual Reference Feedback Tuning (VRFT) [4, 5] and Fictitious Reference Iterative Tuning (FRIT) [6] have attracted attention because controller parameters can be obtained offline from a set of input/output data without repeated experiments. The above control methods, which do not use models of the controlled object, are also being applied to industrial systems [7–10]. VRFT and FRIT have not only been applied to linear systems but they have been extended to nonlinear systems [11–14], including database-driven FRIT [11], VRFT using neural networks [12], VRFT for linear parameter varying systems [13], and VRFT for sparse gain-scheduled PID [14]. However, direct tuning of the LUT parameters using a data-driven control approach has yet to be reported.

Here, we propose a method to directly tune the LUT parameters used in the gain-scheduled control. First, the scheduling function is represented by a LUT, and a gain-scheduled PID controller is defined. Next, the cost function is derived based on the VRFT approach for the LUT-type gain-scheduled PID controller. The cost function has a convex characteristic. Although the least-squares (LS) method is applicable, the number of tuning parameters of a LUT is huge, which may result in overlearning. We propose a cost function to avoid overlearning, where the L_2 norm is introduced to suppress the difference between adjacent LUT parameters. The proposed method eliminates the need for trial-and-error parameter tuning and identification of the system model to be controlled.

The rest of this paper is organized as follows. Section II sets the problem and describes VRFT and gain-scheduled control. Section III details the proposed method to directly tune the LUT parameters based on VRFT. Section IV conducts a simulation example to confirm the effectiveness of the proposed method. The simulation demonstrates that the LUT parameters are optimized and the desired control performance can be obtained. Section V summarizes this study.

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. The name of the corresponding author appears after the financial information, e.g. *(Corresponding author: S. Yahagi).*

S. Yahagi is with the 6th Research Department, ISUZU Advanced Engineering Center Ltd., 8 Tsutidana, Fujisawa-shi, Kanagawa 252-0881, Japan (e-mail: yahagi@iaec.isuzu.co.jp, syahagi@frontier.hokudai.ac.jp).

I. Kajiwara is with Division of Mechanical and Aerospace Engineering, Hokkaido University, N13, W8, Kita-ku, Sapporo, Hokkaido 060-8628, Japan (e-mails: ikajiwara@eng.hokudai.ac.jp).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

II. PRELIMINARY

A. Problem setting

Similar to previous data-driven controls, we consider model-referenced control. Fig. 1 shows a block diagram of model-referenced control with gain-scheduled control. Here, $x \in R^{n_x}$ is the scheduling parameter, which is the measurable state that is inputted into the gain scheduler. $u \in R$ is the control input, $y \in R$ is the output, and $r \in R$ is the set-point. $e \in R$ is the error defined by $e = r - y$. $\rho(t) \in R^m$ is a controller parameter, which is a function of time. $t \in Z$ is the discrete time. $C(z, \rho)$ is the controller given as $u = C(z, \rho)e$. $f(x, w)$ is the scheduling function, which consists of the LUT (described later) that is given as $\rho = f(x, w)$. $w \in R^{n_w}$ is the overall tuning parameter, which constitutes the scheduling function. P is the controlled object, which is often used to validate data-based control [15, 16], and is described as

$$y(t) = f_p(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) \quad (1)$$

where $f_p(\cdot)$ is an unknown nonlinear function. n_u and n_y are unknown orders of the input and output, respectively. We assume that this system is stable and can be linearized at any equilibrium point. The scheduling parameter candidates are associated with the plant output. In general, a nonlinear system must be linearized to design a gain-scheduled controller. This paper aims to design gain-scheduled control without system identification and linearization. We consider the problem of directly tuning the LUT parameter. LUT is used as the gain scheduler and its parameters w are tuned such that the closed-loop characteristics from the set-point value r to the output y match a user-defined reference model M_d. The objective is to optimize the scheduler parameters so that the following cost function is minimized

$$J_{MR}(w) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t,w) - M_d(z)r(t)\right)^2, \quad (2)$$

where $y(t, w)$ is the closed-loop response to $r(t)$ when the controller $C(z, \rho)$ with $\rho = f(x, w)$ is used. N is the data length. Hence, we aim to obtain the optimal parameter w from a set of plant input/output data without using a system model to be controlled. Section IIB describes the structure of the controller. Section IIC details the gain scheduler, which is expressed by a grid-based LUT.

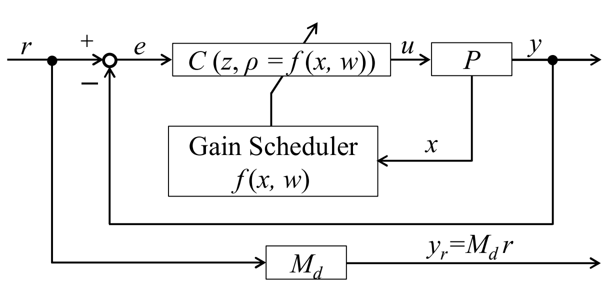

Fig. 1. Gain-scheduled control system with a reference model.

B. Gain-scheduled controller

The gain-scheduled controller shown in Fig. 1 can be described as

$$C(z, \rho) = \rho^T(t)\beta(z) = f(x, w)^T\beta(z) \quad (3)$$

with

$$\begin{aligned} \rho(t) &= [\rho_1(t) \quad \rho_2(t) \quad \cdots \quad \rho_m(t)]^T \\ \beta(z) &= [\beta_1(z) \quad \beta_2(z) \quad \cdots \quad \beta_m(z)]^T \\ f(x, w) &= [f_1(x, w^{\rho_1}) \quad f_2(x, w^{\rho_2}) \quad \cdots \quad f_m(x, w^{\rho_m})]^T \\ w &= [(w^{\rho_1})^T \quad (w^{\rho_2})^T \quad \cdots \quad (w^{\rho_m})^T]^T, \end{aligned} \quad (4)$$

where $\rho_j = f_j(x, w^{\rho_j})$, $\rho_j \in R$ is the controller parameter, $\beta_j(z)$ is a scalar rational function, $f_j: R^{n_x} \times R^M \to R$ is the scheduling function for ρ_j, and $w^{\rho_j} \in R^M$ is an M-dimensional vector of the LUT tuning parameter for ρ_j, $j \in \{1, 2, \dots, m\}$. $w \in R^{n_w}$, which is the overall LUT parameter, is the n_w-dimensional vector consisting of w^{ρ_j}. The length of w is given by $n_w = m \times M$. ρ_j is gain-scheduled according to x because w^{ρ_j} is fixed after tuning. ρ and $\beta(z)$ are m-dimensional vectors consisting of ρ_j and $\beta_j(z)$, respectively. $f: R^{n_x} \times R^{n_w} \to R^m$ is a vector-valued function consisting of f_j. We focus on the velocity form of the PID controller (Fig. 2) because it is most common in industry and is compatible with gain-scheduled control [14]. Here, $\rho_p(t)$, $\rho_i(t)$, and $\rho_d(t)$ are the proportional, integral, and derivative gains, respectively; Δ represents $1 - z^{-1}$; z^{-1} is the backward operator. Thus, the gain-scheduled PID control is represented by

$$\Delta u(t) = C_v(z, \rho)e(t) \quad (5)$$

with

$$C_v(z, \rho) = \rho^T(t)\beta(z) \quad (6)$$

$$\rho(t) = [\rho_p(t) \quad \rho_i(t) \quad \rho_d(t)]^T \quad (7)$$

$$f(x, w) = [f_p(x, w^{\rho_p}) \quad f_i(x, w^{\rho_i}) \quad f_d(x, w^{\rho_d})]^T \quad (8)$$

$$\beta(z) = [1 - z^{-1} \quad 1 \quad (1 - z^{-1})^2]^T, \quad (9)$$

where $f_j (j = p, i, d)$ is a LUT for proportional, integral, and derivative gains, respectively

C. Grid-Based LUT [17, 18]

In industry, the LUT-based gain scheduler is often adopted for nonlinear systems because it is intuitive and easy to understand [2, 3]. We explain a LUT f_j for ρ_j. In this section, ρ_j, w^{ρ_j}, and f_j are denoted as ρ, w, and f, respectively, to avoid complications. Fig. 3 overviews a two-dimensional grid-based LUT. Fig. 3 (a) shows a LUT's overview. The LUT size is $M_1 \times M_2$. M_1 and M_2 represent the number of interpolation nodes in each axial direction. x_1 and x_2 are the user-defined inputs to a LUT. $c_{1,1}, \dots, c_{1,M_1}$ and $c_{2,1}, \dots, c_{2,M_2}$ are the user-defined interpolation nodes. $\theta_{k,l} \in R$ ($k = 1$–M_1, $l = 1$–M_2) are the weight coefficients representing the heights of the interpolation nodes and are the LUT parameters

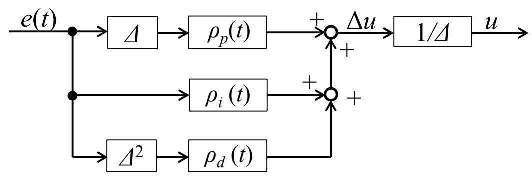

Fig. 2. Block diagram of the velocity form of the PID controller.

to be tuned. $\theta \in R^{M_1 \times M_2}$ is the matrix consisting of $\theta_{k,l}$. Fig. 3 (b) (c) shows the area after determining the four surrounding nodes of the current operating point. $x \in R^2$ is a vector consisting of x_1 and x_2. The LUT's output ρ after determining the four surrounding nodes of the current operating point is calculated by

$$\begin{aligned}\rho = f(x) = \theta_{k,l}\frac{A_{k+1,l+1}}{A} + \theta_{k+1,l}\frac{A_{k,l+1}}{A}\\ +\theta_{k,l+1}\frac{A_{k+1,l}}{A} + \theta_{k+1,l+1}\frac{A_{k,l}}{A}\end{aligned} \tag{10}$$

with

$$\begin{aligned}A_{k+1,l+1} &= (c_{1,k+1} - x_1)(c_{2,l+1} - x_2)\\ A_{k,l+1} &= (x_1 - c_{1,k})(c_{2,l+1} - x_2)\\ A_{k+1,l} &= (c_{1,k+1} - x_1)(x_2 - c_{2,l})\\ A_{k,l} &= (x_1 - c_{1,k})(x_2 - c_{2,l})\\ A &= (c_{1,k+1} - c_{1,k})(c_{2,l+1} - c_{2,l}).\end{aligned} \tag{11}$$

Next, the basis function $\phi_{k,l}(x, c)$ is introduced to obtain the generalized expression for (10). This function finds the four nodes surrounding the current operating point and computes the corresponding normalized region using (11) [17, 18]. Considering the basis function, the generalized LUT's output is expressed as

$$\rho = f(x) = \sum_{k=1}^{M_1}\sum_{l=1}^{M_2}\theta_{k,l}\phi_{k,l}(x, c)\,. \tag{12}$$

Additionally, (12) can be described as

$$\rho = f(x) = w^T\tilde{\phi}(x, c) = \sum_{i=1}^{M_1M_2} w_i\,\tilde{\phi}_i(x, c) \tag{13}$$

with

$$\begin{aligned}w &= [w_1 \quad w_2 \quad \cdots \quad w_{M_1M_2}]^T\\ &= [\theta_{1,1} \quad \theta_{1,2} \quad \cdots \quad \theta_{M_1,M_2}]^T\end{aligned} \tag{14}$$

$$\begin{aligned}\tilde{\phi} &= [\tilde{\phi}_1 \quad \tilde{\phi}_2 \quad \cdots \quad \tilde{\phi}_{M_1M_2}]^T\\ &= [\phi_{1,1} \quad \phi_{1,2} \quad \cdots \quad \phi_{M_1,M_2}]^T,\end{aligned} \tag{15}$$

where the matrix elements are converted to vector elements using the following equations:

$$w_{k+(l-1)M_1} = \theta_{k,l}, \tilde{\phi}_{k+(l-1)M_1} = \phi_{k,l}. \tag{16}$$

It should be noted that w and θ both represent LUT parameters, but they are expressed as a vector and matrix, respectively. Similarly, $\tilde{\phi}$ and ϕ differ only in their expressions. Below, expression (13) is used because it can be easily handled by mathematical expressions.

III. DATA-DRIVEN TUNING OF THE LUT PARAMETERS

A. VRFT

The standard VRFT [4, 5] directly tunes the control parameters from open-loop input/output data without system model. The controller parameters are tuned so that the reference model and the closed-loop system have the same characteristics. We consider the fixed-order LTI controller $C_{VR}(\rho, z)$ which is parameterized through the controller parameters ρ. The plant is a stable system. The VRFT procedure is as follows:

[Step 1] Acquire the initial plant input/output data $D = \{u_0(t), y_0(t) | t = 1, \dots N\}$ in a test. The user sets the reference model M_d, which is the desired closed-loop system.

[Step 2] Considering $y_0(t)$ to be the output of the reference model, express the virtual reference signal

$$\bar{r}(t) = M_d^{-1}y_0(t). \tag{17}$$

[Step 3] Considering $\bar{r}(t)$ to be the reference input of the closed-loop system, describe the virtual control input as

$$\bar{u}(t) = C_{VR}(\rho, z)\big(\bar{r}(t) - y_0(t)\big). \tag{18}$$

[Step 4] If the virtual control input and the acquired initial input data are close, regard the closed-loop system as close to the reference model. Thus, the cost function is given as

$$J_{VR}(\rho) = \frac{1}{N}\sum_{t=1}^{N}\big(u_0(t) - \bar{u}(t)\big)^2\,. \tag{19}$$

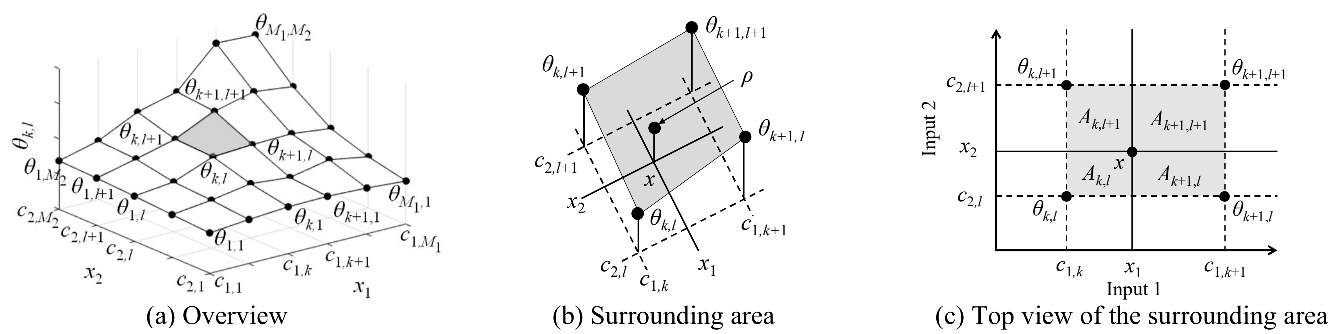

Fig. 3. Two-dimensional look-up table [17, 18].

[Step 5] Introduce a prefilter L. The prefilter is effective when the inverse of the reference model is nonproper or when the chosen controller structure does not belong to the class of controllers that can achieve the reference closed-loop model [4, 5, 14]. The cost function with a prefilter is described as

$$J_{VR}(\rho) = \frac{1}{N}\sum_{t=1}^{N}\left(u_L(t) - C(\rho, z)e_L(t)\right)^2 \tag{20}$$

with

$$u_L(t) = Lu_0(t), e_L(t) = L\bar{e}(t). \tag{21}$$

B. Derivation of the cost function

Based on (13), the PID gains ρ_j $(j = p, i, d)$ are computed as

$$\rho_j = f_j(x) = (w^{\rho_j})^T \tilde{\phi}(x, c) = \sum_{i=1}^{M_1 M_2} w_i^{\rho_j}\ \tilde{\phi}_i(x, c) \tag{22}$$

with

$$w^{\rho_j} = \begin{bmatrix} w_1^{\rho_j} & w_2^{\rho_j} & \dots & w_{M_1 M_2}^{\rho_j} \end{bmatrix}^T \tag{23}$$
$$\tilde{\phi} = \begin{bmatrix} \tilde{\phi}_1 & \tilde{\phi}_2 & \dots & \tilde{\phi}_{M_1 M_2} \end{bmatrix}^T, \tag{24}$$

where $w^{\rho_j}(j = p, i, d)$ is the LUT parameter for proportional, integral, and derivative gains, respectively. From the VRFT framework and the LUT structure, the cost function for the optimized LUT parameters is given by

$$J_{LUT}(w) = \frac{1}{N}\sum_{t=1}^{N}\left(d(t) - w^T\xi(t)\right)^2 \tag{25}$$

with

$$d = L\Delta u(t) \tag{26}$$
$$w = [(w^{\rho_p})^T \quad (w^{\rho_i})^T \quad (w^{\rho_d})^T]^T \tag{27}$$
$$\xi(t) = \begin{bmatrix} \tilde{\phi}(x, c)\beta_1(z) \\ \tilde{\phi}(x, c)\beta_2(z) \\ \tilde{\phi}(x, c)\beta_3(z) \end{bmatrix} e_L(t). \tag{28}$$

Since the cost function is convex, the LS method yields the optimal solution as

$$w^* = (Z^T Z)^{-1} Z^T D \tag{29}$$

with

$$Z = [\xi(1) \quad \xi(2) \quad \cdots \quad \xi(N)]^T \tag{30}$$
$$D = [d(1) \quad d(2) \quad \cdots \quad d(N)]^T. \tag{31}$$

Applying the LS method gives the LUT parameters, but the difference between neighboring LUT parameters may become extremely large. This leads to sudden changes in the PID gain and instability in the closed-loop system. To suppress the significant gain fluctuation, the L_2 norm is introduced for the difference between LUT adjacent parameters. The cost function, including the L_2 norm, is expressed as

$$J_{LUT}(w) = \|D - Z^T w\|_2^2 + \lambda\|\psi w\|_2^2, \tag{32}$$

where λ is a positive constant and a design parameter that tunes the relative strength between the sum of the squared error terms and the regularization term. Here, cross-validation provides an appropriate value λ. The optimization problem of the cost function (32) is called the generalized ridge regression. When ψ is an identity matrix, (32) is called the ridge regression. The second term of (32) is called the total variation [19]. It represents the difference between parameters in close proximity, and is expressed as

$$\begin{aligned}\|\psi w\|_2^2 = \Bigg(&\sum_{i=1}^{M_1 M_2 - 1} |w_{i+1} - w_i|_2^2 \\ &- \sum_{i=1}^{M_2 - 1} |w_{i\times M_1 + 1} - w_{i\times M_1}|_2^2\Bigg) \\ &+ \sum_{i=1}^{M_1 M_2 - M_1} |w_{i+M_1} - w_i|_2^2 .\end{aligned} \tag{33}$$

The first two terms represent the first axis direction, while the third term represents the second axis direction. ψ can be expressed by the matrix. The optimal solution of cost function (32) can be analytically obtained as

$$w^* = (Z^T Z + \lambda\psi^T\psi)^{-1} Z^T D. \tag{34}$$

Herein k-fold cross-validation is used to directly determine the optimal regularization parameter λ from multiple λ. For the details, refer to the literature [20].

C. Algorithm

The algorithm for the direct data-driven tuning of the LUT parameters employs the following steps:

[Step 1] Measure input/output data $D = \{u_0(t), y_0(t) | t = 1, \dots N\}$ in an open-loop test.

[Step 2] Determine the input signals to LUTs (the scheduling parameters) x_1, x_2. Set the sizes M_1, M_2 and interpolation nodes c_1, c_2 of LUTs (see Fig. 3). Additionally, the LUT size is determined based on the maximum and minimum values of the input signals to LUTs. Set the reference model M_d and prefilter L. Similar to previous studies [5, 10, 14], the prefilter used in this paper is M_d.

[Step 3] After calculating (30) and (31) using (26) and (28), respectively, minimize the cost function (32) using (34) and the cross-validation to find the weight coefficients (LUT parameters) of the scheduling function. The proposed method can be implemented from the above algorithm. In addition, the implementation of the controller uses the LUT-type gain-scheduled PID controller, as shown in (5)–(9), and the PID gains are calculated using (22).

Remark 1. The previous study [11] for nonlinear system is based on FRIT and requires nonlinear optimization, which is computationally expensive. The obtained optimal value may fall into a local solution. On the other hand, the cost function in the proposed method is convex with respect to the tuning parameters. Thus, the solution can be uniquely determined using the LS method for numerous tuning parameters of LUT, which does not require much computation time.

Remark 2. In gain-scheduled control, abrupt gain fluctuations are undesirable because they destabilize the system. The proposed method can suppress sudden gain changes by introducing a cost function, which includes the L_2 norm to suppress the difference between adjacent parameters. Furthermore, cross-validation can automatically obtain the design parameter λ.

Remark 3. Since the proposed algorithm is based on VRFT, the stability of the closed-loop system is not guaranteed. This is a drawback of the proposed method. However, this is a common problem in direct data-driven control, and it remains an attractive approach when the system is complex or models are unavailable due to development costs (engineering time and required hardware) [21]. In the proposed method, the generalized ridge regression can suppress the closed-loop instability.

IV. Numerical Example

A. System Description

The controlled object is the Hammerstein model, which is often used to describe nonlinear systems [22]. It has also been used as a verification model for data-driven control [11, 14, 15]. The plant is described as

$$\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) + 1.2v(t-1) \\ &\quad -0.1v(t-2) + \xi(t) \\ v(t) &= 1.5u(t) - 1.5u^2(t) + 0.5u^3(t), \end{aligned} \tag{35}$$

where ξ is white noise with variance 1×10^{-3}. The reference model [11, 14], which has a time constant of 2 s, is given as

$$M_d(z^{-1}) = \frac{0.399z^{-1}}{1-0.736z^{-1}+0.135z^{-2}}, \tag{36}$$

The inputs to LUT (scheduling parameters) are given by

$$x_1(t) = y(t), x_2(t) = y(t)(1-z^{-1}). \tag{37}$$

The sampling period is set to 1 s. These system settings are the same as the literature [11, 14].

B. Simulation Results and Discussion

The results are described based on the algorithm shown in Section IIIC. In Step 1, a chirp sine signal (frequency 0–0.5 Hz, amplitude 1.75, offset 1) is applied to the input. The frequency band is determined based on the time constant of the reference model and the sampling period. Then the input/output data are measured. Fig. 4 shows the initial input/output data obtained by the open-loop test. A total of 800 data points are collected. The number of the fold for the cross-validation is set to 2. In Step 2, the grid spacing of LUT is set to 1.0, and the LUT size is defined as $M_1 = 13$ (range: -5 to 7) and $M_2 = 11$ (range: -5 to 5). That is, $c_1 = [-5, -4, \dots, 6, 7]$ and $c_2 = [-5, -4, \dots, 4, 5]$. The input signals to the LUTs are shown in (37). Similar to other VRFT studies [5, 10, 14], the prefilter L uses M_d. In Step 3, the LUT parameters are tuned from the data. Fig. 5 shows the tuned LUTs for the PID gain by LUT-VRFT-Ridge. The obtained LUTs consist of x_1-, x_2-, and $w^{\rho_j}(j = p, i, d)$-axes, where w^{ρ_j} is the LUT parameter for each PID gain. The lengths of w^{ρ_j} and w are 143 ($= M_1M_2$) and 429 ($= 3M_1M_2$), respectively. The obtained λ is 0.5. We can confirm that many parameters are tuned. Finally, the controller is implemented. The PID gains are adapted according to x using the obtained LUTs (see (22)). Fig. 6 shows the time series data after the LUT parameters are tuned from the measured input/output data. For comparison, the time series data are shown when using the CHR method, which is a classical PID gain tuning method, the standard VRFT (fixed PID gain), VRFT for LUT applying the LS method (LUT-VRFT-LS), and VRFT for LUT with the generalized ridge (LUT-VRFT-Ridge). As reported previously, the PID gains obtained by the CHR method were $K_p = 0.059$, $K_i = 0.058$, and $K_d = 0.0038$ [11, 15]. The fixed PID gains obtained by VRFT were $K_p = 0.0862$, $K_i = 0.1597$, and $K_d = 0.0037$. From the top, the output, input, proportional gain, integral gain, and derivative gain are shown. Table I shows the value of the cost function (2), which indicates the tracking error performance, for CHR, VRFT, LUT-VRFT-LS, and LUT-VRFT-Ridge. The response was very slow in the CHR method. A comparison with the standard VRFT and LUT-VRFT-LS confirmed that LUT-VRFT-LS had a PID gain, which changed based on the scheduling parameters. The response with LUT-VRFT-LS followed the desired response. Comparing LUT-VRFT-LS and LUT-VRFT-Ridge demonstrated that the overshoot in LUT-VRFT-Ridge was smaller than that in LUT-VRFT-LS, and the PID gain in LUT-VRFT-Ridge had a smoother change than that in LUT-VRFT-LS. Next, we checked the effect of lattice spacing of LUTs. Table II shows that the closed-loop stability depended on the lattice spacing. The system in the LS case became unstable as the lattice spacing decreased. This was attributed to an increase in the LUT parameters, which led to overlearning. On the other hand, the closed-loop system became stable when the generalized ridge generated optimal parameters. These results confirm the effectiveness of the proposed method.

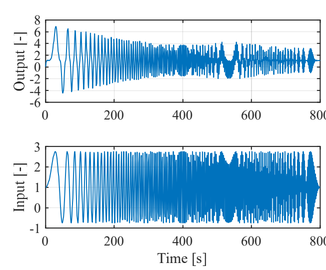

Fig. 4. Time series data under an open-loop condition.

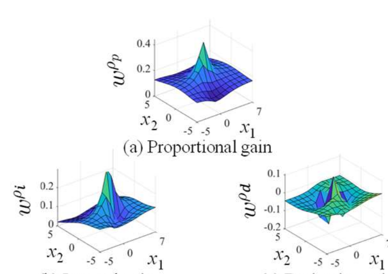

Fig. 5. Tuned LUTs by the proposed method.

TABLE I. RESULTS OF THE TRACKING ERROR

Fixed (CHR)	1.548×10^{-1}
Fixed (VRFT)	5.562×10^{-2}
LUT-VRFT-LS	2.557×10^{-2}
LUT-VRFT-Ridge	1.173×10^{-2}

TABLE II. CLOSED-LOOP STABILITY

Interval	LS	Ridge
2.0	Stable	Stable
1.0	Stable	Stable
0.5	Unstable	Stable

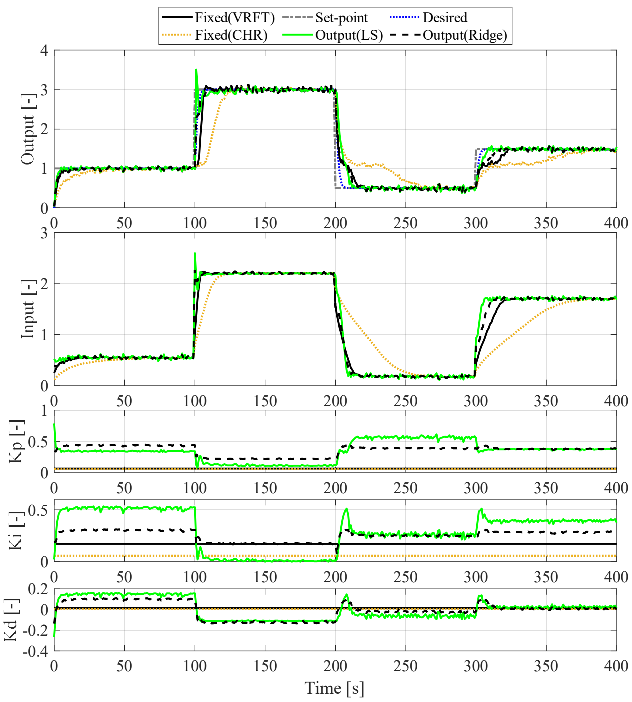

Fig. 6. Time series data with the proposed method.

V. CONCLUSION

Here, we proposed a direct data-driven design for the gain-scheduled controller, which consists of LUTs. First, a cost function was derived based on VRFT to tune LUTs. The optimized parameters were obtained by the LS method. Next, the L_2 norm of the magnitude of the difference between adjacent LUT parameters was introduced to prevent overlearning. The proposed method enabled LUT parameter tuning from plant input/output data without a system model. Moreover, the analytical solution could be obtained using the generalized ridge regression. A numerical example demonstrated that the proposed method could generate numerous LUT parameters without knowledge of the controlled object's characteristics. Moreover, using the cost function with the L_2 norm avoided overlearning. Consequently, the proposed method eliminates the need for trial-and-error LUT parameter tuning. In the future, we will analyze for noise, compare LUTs obtained by the proposed method and a model-based approach, and apply the proposed approach to industrial systems such as automobiles.

REFERENCES

[1] K. Åström and T. Hägglund, "The future of PID control, Control Engineering Practice," Control Engineering Practice, vol. 9, no. 11, pp. 1163–1175, 2001.

[2] E. B. Ondes, I. Bayezit, I. Poergye, A. Hafsi, "Model-based 2-D look-up table calibration tool development," in Proceedings of the 11th Asian Control Conference (ASCC), Gold Coast, QLD, Australia, 2017, pp.1011–1016.

[3] C. Guardiola, B. Pla, D. Blanco-Rodriguez, P. Cabrera, "A learning algorithm concept for updating look-up tables for automotive applications," Mathematical and Computer Modelling, vol. 57, no. 7, pp. 1979–1989, 2013.

[4] M. C. Campi, A. Lecchini and S. M. Savaresi, "Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers," Automatica, vol. 38, no. 8, pp. 1137–1146, 2002.

[5] S. Formentin, S. M. Savaresi and L. D. Re, "Non-iterative direct data-driven controller tuning for multivariable systems theory and application," IET Control Theory and Applications, vol. 6, no. 9, pp. 1250-1257, 2012.

[6] S. Soma and O. Kaneko, "A new method of controller parameter tuning based on input-output data–Fictitious Reference Iterative Tuning (FRIT)," IFAC Proceedings Volumes, vol. 37, no. 12, pp. 789–794, 2004.

[7] S. Formentin, G. Panzani and S. M. Savaresi, "VRFT for LPV systems: Theory and braking control application," Robust Control and LPV Approaches, pp. 289–309, 2013.

[8] T. E. Passenbrunner, S. Formentin, S. M. Savaresi and L. Re, "Direct multivariable controller tuning for internal combustion engine test benches," Control Engineering Practice, vol. 29, pp. 115–122, 2014.

[9] S. Yahagi, I. Kajiwara and T. Shimozawa, "Slip control during inertia phase of clutch-to-clutch shift using model-free self-tuning proportional-integral-derivative control," Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 234, no. 9, pp. 2279–2290, 2020.

[10] S. Yahagi and I. Kajiwara, "Direct tuning of gain-scheduled controller for electro-pneumatic clutch control," Advanced in Mechanical Engineering, vol. 13, no. 7, pp. 1–12, 2021.

[11] S. Wakitani, Y. Ohnishi and T. Yamamoto, "Design of FRIT-based nonlinear PID control systems," Journal of The Society of Instrument and Control Engineers, vol. 52, no. 10, pp. 885–891, 2013.

[12] P. Yan, D. Liu, D. Wang, and H. Ma, "Data-driven controller design for general MIMO nonlinear systems via virtual reference feedback tuning and neural networks, " Neurocomputing, vol. 171, 815-825, 2016.

[13] S. Formentin, D. Piga, R. Toth and S. M. Savaresi, "Direct learning of LPV controllers from data," Automatica, vol. 65, pp. 98–110, 2016.

[14] S. Yahagi and I. Kajiwara, "Direct Tuning Method of Gain-Scheduled Controllers with the Sparse Polynomials Function," Asian journal of Control, pp. 1–16, 2021. Online published.

[15] T. Yamamoto, K. Takao and Y. T, "Design of a Data-Driven PID Controller," IEEE Transactions on Control Systems Technology, vol. 17, no. 1, pp. 29–39, 2009.

[16] S. Stenman, F. Gustafsson and L. Ljung, "Just in time models for dynamical systems," in Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 13–13 Dec. 1996.

[17] M. Vogt, R.B. Gmbh, R. Isermann, "On-Line Adaptation of Grid-Based Look-up Tables Using a Fast Linear Regression Technique," Journal of Dynamic Systems Measurement and Control, vol. 126, no. 4, pp. 732–739, 2004.

[18] F. Hausberg, M. Plöchl, M. Rupp, P. Pfeffer, S. Hecker, "Combination of map-based and adaptive feedforward control algorithms for active engine mounts," Journal of Vibration and Control, vol. 23, no. 19, pp. 3092-3107, 2016.

[19] M. Nagahara, Sparsity Methods for Systems and Control, Now Publishers, 2020.

[20] M. Bishop C, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, New York, 2006.

[21] P. Polack, S. Delprat, B d'Andréa-Novel, "Brake and velocity model-free control on an actual vehicle," Control Engineering Practice, vol. 92, no.8, 104072, 2019.

[22] Z. Lang, "On identification of the controlled plants described by the Hammerstein model," IEEE Transactions on Automatic Control, vol. 39, no. 3, pp. 569–573, 1994.