Descriptive title:
Direct Tuning Method of Gain-Scheduled Controllers with the Sparse Polynomials Function

Concise title:
Direct Tuning Method of Gain-Scheduled Controllers

Shuichi Yahagi (corresponding author)
Affiliations and full mailing address:
Division of Human Mechanical Systems and Design, Hokkaido University, N13, W8, Kita-ku, Sapporo, Hokkaido 060-8628, Japan.
6th Research Department, ISUZU Advanced Engineering Center Ltd., 8 Tsutidana, Fujisawa-shi, Kanagawa 252-0881, Japan.

E-mail: yahagi@iaec.isuzu.co.jp, syahagi@eis.hokudai.ac.jp

Itsuro Kajiwara
Affiliations and full mailing address:
Division of Human Mechanical Systems and Design, Hokkaido University, N13, W8, Kita-ku, Sapporo, Hokkaido 060-8628, Japan.

E-mail: ikajiwara@eng.hokudai.ac.jp

Direct Tuning Method of Gain-Scheduled Controllers with the Sparse Polynomials Function

Keywords:
Data-driven control, VRFT, Gain-scheduled control, Model-free design, LASSO regression, Sparse

Abstract:
In industry, gain-scheduled PID control is performed for nonlinear systems using a look-up table (LUT) that is easy to understand. Compared with the fixed PID, there are many more parameters of the scheduler, and it takes a lot of time to tune them. Also, the ROM storage area increases. To address these problems, in this paper, we propose a gain-scheduled control law using the sparse polynomial functions and a direct parameter tuning method without system identification. The polynomial functions are used instead of LUT to reduce the ROM area. For direct tuning, data-driven control is formulated so that it can be applied to the gain-scheduled control, and the optimal parameters are obtained by the LASSO regression, with which the small contributing parameters of the scheduler become zero, and a sparse controller is obtained. The effectiveness of this method was examined by simulation for two types of nonlinear systems. As a result, it was revealed that a sparse controller with a low calculation cost and a reduced ROM area can be directly obtained without knowing the characteristics of the controlled object for a large number of control parameters of the gain scheduler.

1. Introduction

Over 90% of the closed-loop control methods in industrial systems use PID (Proportional-Integral-Derivative) control because the perspective is easy to understand and the calculation cost is low [1]. Also, the desired control performance can be obtained if the controlled object has strong linearity, but in the case of a nonlinear system, it is difficult to obtain sufficient control performance with PID control with fixed gains [2]. The nonlinear control theory and robust control can be applied, but the hurdles of nonlinear control applications are high because the theory is complicated and the calculation load is often large. In addition, it is generally difficult to address a robust controller for a system whose characteristics significantly change. Against this background, the gain-scheduled control is one of the most popular approaches to nonlinear control design [3, 4], and it is well-known as an effective and economical method for actual nonlinear control designs [5]. Also, the gain-scheduled control is a method that realizes the desired control performance by changing the controller parameters according to the state of the controlled object and the external environment. This approach is intuitively understandable and easy to accept in many industries. Also, the gain-scheduled PID control, which uses a look-up table (LUT), is often used in many industries [6, 7]. At the same time, a large number of control parameters must be tuned to obtain the desired control performance. The fixed PID control has three tuning parameters, but the gain-scheduled control with LUT has a much greater number of parameters that must be tuned. Hence, it takes a lot of time for parameter tuning. The model-based design of the LPV (Linear Parameter-Varying) controller has been proposed, but it may still be difficult to obtain highly accurate models because industrial systems are often complicated [8], so the controller may not be fully performed.

In recent years, the control system design methods that do not need controlled object models and system identification have attracted a lot of attention. The data-driven control, such as VRFT (Virtual Reference Feedback Tuning) [9] and FRIT (Fictitious Reference Feedback Tuning) [10, 11], is drawing a lot of attention because the controller parameters can be obtained off-line from a set of input/output data without repeated experiments. As model-free control methods for sequentially changing controller parameters online, the adaptive PID control with SPSA (Simultaneous Perturbation Stochastic Approximation Algorithm) [12], adaptive PID [13, 14], and FRIT-RLS [15, 16], which is an online version of the FRIT, and so on, have been proposed. It is noteworthy that these model-free controls can be applied to time-varying systems because the control parameters can be adapted in real time. The above-mentioned data-driven control and model-free control methods, which do not use the model of the controlled object, are currently applied to industrial systems, such as process systems, automobile systems, and vibration control problems [17–22].

In general, it is possible to apply online model-free control to nonlinear and time-varying systems, but in the case of systems with fast parameter fluctuations, it is necessary to quickly update the parameters, which results in increasing the computational load. When considering the implementation on mass products, it is difficult to guarantee the performance and safety of the shipped products to the market unless stability is theoretically guaranteed. Then, we preferred to use the predesigned controller parameters according to the state and environment of the controlled object in advance rather than those estimated online. Therefore, we considered that the data-driven control, which can tune the parameters off-line, is an effective method. Meanwhile, most of the data-driven control is based on a linear control system. Since many complex nonlinear systems exist in industrial systems, a data-driven control system that realizes nonlinear control is required. There are applications to nonlinear systems, such as DD-PID [23] and DD-FRIT [24, 25], FRIT (using feedback linearization) [26], and VRFT (for LPV systems) [27–29]. The DD-PID and DD-FRIT have relatively high storage capacity and computational cost when a controller is implemented. In FRIT, with feedback linearization, it is necessary to know the model structure in advance, and it is not possible to directly obtain the PID controller parameters, which are mostly used in many industries. In the VRFT for the LPV system, the gain-scheduled controller is obtained, which has a small calculation cost and storage capacity at implementation. Also, it is possible to interpret and understand the same, as the gain-scheduled PID control uses the described LUT at the beginning by visualizing the schedulers for each of the obtained PID gains. Therefore, it is desirable to obtain the parameters for the gain-scheduled PID control by the VRFT.

In the gain-scheduled control, the selection process of the scheduling parameters is also important. For example, the position control for the spring-mass system is very important, and it is combined with different states, such as the position, velocity, or acceleration. If the selected scheduling parameters do not affect the change in the characteristics of the controlled object, the calculation cost and ROM area would increase for unnecessary parameters. In the regression problem of machine learning, to prevent overlearning, a method called LASSO (least absolute shrinkage and selection operator) regularization, in which the L_1 norm is added to the cost function is performed. Mainly, this is a method for extracting essential low-dimensional information with high accuracy from high-dimensional information. In the model obtained by LASSO, the elements with little information representing the feature are zero, and such a property is called sparseness.

In this paper, the authors propose a method that directly tunes the parameters of the scheduling function (scheduler) for gain-scheduled PID control while considering the sparseness property of LASSO. The gain-scheduled PID controller is defined by expressing the scheduling function as a polynomial that consists of weight coefficients, which are the tuning parameters. By using a polynomial instead of the LUT, the tuning parameters can be reduced. Next, the cost function of the VRFT for the gain-scheduled PID controller is derived. Also, the value of the optimum control parameters is calculated by LASSO so that the cost function becomes minimum. In addition, the gain scheduler with the high sparseness can be constructed by LASSO. As a result, trial and error parameter tuning and system identification are not needed. Moreover, the ROM area of the controller and the calculation cost are reduced. In other words, the controller with the high sparseness and reduced parameter tuning man-hours can be obtained. Studies [27–29] for applying the VRFT to the LPV system have been previously proposed. In comparison with those studies, the main features of this study are as follows. For industrial applications, the PID controller is assumed as a control law, the gain scheduler adopts a quadratic polynomial, and the sparse controller can be obtained.

The structure of this paper is as follows. In Chapter 2, the problem formulation and VRFT are described as preliminary. Section 3 describes the proposed method and the direct tuning method of the gain-scheduled control parameters using the VRFT. In Section 4, two types of nonlinear systems are controlled using the proposed method by numerical simulation, the correspondence of the controller parameters to the characteristic fluctuations is revealed, and the desired control performances are obtained. Chapter 5 provides a summary of this paper.

2. Preliminary

The problem formulation and VRFT are explained. The VRFT is one of the data-driven controls that are used in this paper.

2.1. Problem formulation

When the controlled object has complicated characteristics, such as nonlinearity or time-varying, the performance deterioration cannot be avoided with a time-invariant (fixed) controller, whereas a gain-scheduled controller can be an effective method for realizing good control performance. The gain-scheduled control system shown in Figure 1 is considered the control law. In the figure, $u \in R$ is the control input, $y \in R$ is the output, r is the set-point, e is the error, and P is the controlled object. The gain-scheduled control is constructed from the controller $C(z, \rho)$, the controller parameter vector ρ, the rational function vector $\psi(z)$, the scheduling parameter $\theta \in \Theta$, the parameter vector w, and the scheduling function (scheduler) $f(\theta, w)$. The controller is described as

$$C(z,\rho) = \rho^T \psi(z) = f(\theta)^T \psi(z), \tag{1}$$

where

$$
\begin{gathered}
\rho = [\rho_1 \quad \rho_2 \quad \cdots \quad \rho_n]^T, \\
f(\theta, w) = [f_1(\theta, w_1) \quad f_2(\theta, w_2) \quad \cdots \quad f_n(\theta, w_n)]^T, \\
\psi(z) = [\psi_1(z) \quad \psi_2(z) \quad \cdots \quad \psi_n(z)]^T, \\
w = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T,
\end{gathered} \tag{2}
$$

where z is the shift operator, and w_i is a parameter vector that constructs the i-th scheduling function $f_i(\theta, w_i)$. The controller parameter vector ρ changes according to the scheduling function $f(\theta, w)$. The detailed settings of the rational function vector $\psi(z)$ are described in Section 3.1.1.

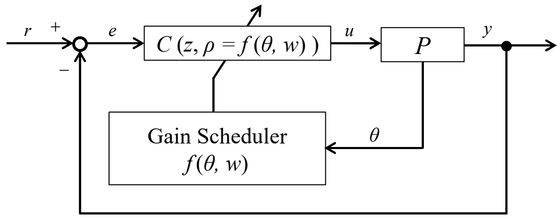

Figure 1 Gain-scheduled control system

We considered the model reference control as well as the previous studies of the data-driven control [9, 11]. Figure 2 shows a block diagram of the model reference gain-scheduled control. The problem setting is to acquire the sparse controller by automatically tuning the control parameters of the gain scheduler so that the transfer characteristics from the set-point to the output can match the reference model M_d, which is determined by the designer. In other words, our aim is to obtain the optimum parameters that make up the sparse gain-scheduled controller that minimizes the following cost function.

$$
J_{MR}(w) = \|y(t, w) - M_d(z) r(t)\|_2^2 + \lambda \|w\|_1, \tag{3}
$$

where λ denotes the weight coefficients, and the second term is the weight-related to w. The details are provided in Section 3.3.

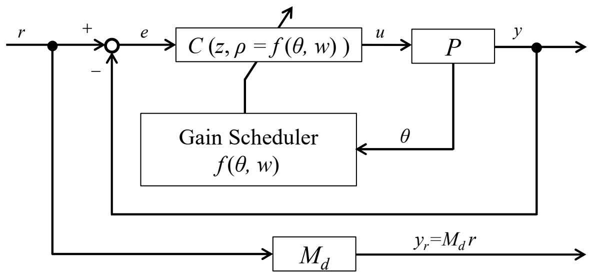

Figure 2 Gain-scheduled control system and reference model

2.2. Controlled object

The LPV system whose characteristics are changed by the scheduling parameters was used in many previous studies. However, the target system is unknown if it is strictly the LPV system or not. Hence, the controlled objects are unknown two-type nonlinear single-input single-output (SISO) systems, which are an explicit LPV system and a system that is not explicitly described as an LPV system. The first system, the LPV system, is described as

$$\begin{cases} x(t+1) = A_d(\theta)x(t) + B_d(\theta)u(t) \\ \qquad\quad y(t) = C_d(\theta)x(t) \end{cases}, \tag{4}$$

where $u \in R$ is the control input, $y \in R$ is the output, $x \in R^n$ is the state, and θ is the scheduling parameter. The matrices A_d, B_d, and C_d are bounded and unknown functions that are continuous with respect to θ, and the candidates of the scheduling parameters are associated with the controlled object output or state. We assumed that this system is stable. The second system is often used in the validation of the database control [23, 30, 31] as

$$y(t) = f_p\big(y(t-1), \dots, y(t-n_y), \\ u(t-1), \dots, u(t-n_u)\big), \tag{5}$$

where $f_p(\cdot)$ is an unknown nonlinear function, and n_u and n_y denote the unknown orders of the input and output, respectively. We assumed that this system is stable and can be linearized at an equilibrium point. The candidates of the scheduling parameters are associated with the controlled object output. Thus, in general, it is necessary to linearize the nonlinear system to design a gain scheduler, but this study aims at designing the gain scheduler without system identification and linearization.

2.3. Standard VRFT [9]

The VRFT is a method for directly tuning the control parameters for LTI (linear time-invariant) systems from open-loop input/output data without system identification. The optimal control parameters are tuned so that the reference model and closed-loop system have the same characteristics. Figure 3 shows the structure of the VRFT. Here, C_{VR} (ρ, z), M_d, and P_{VR} represent the fixed-order controller, which is parameterized through ρ, the reference model, and the controlled object (plant), which is the stable LTI system, respectively. $u(t)$ and $y(t)$ denote the input and output, respectively. ρ denotes the controller parameters. $\bar{r}(t)$ and $\bar{e}(t)$ are the proposed virtual reference input and virtual error in the VRFT, respectively. The VRFT procedure is briefly described as follows:

[Step 1] The input and output data of the plant $u(t)$, $y(t)$, $t = 1, \dots, N$ is acquired in an open-loop test.

[Step 2] The reference model M_d, which is the desired closed-loop, is set.

[Step 3] Considering $y(t)$ as the output of the reference model, the virtual reference input, which generates $y(t)$, can be expressed as

$$\bar{r}(t) = M_d^{-1} y(t). \tag{6}$$

[Step 4] Considering $\bar{r}(t)$ as the reference input of the closed-loop in Figure 3, the virtual control input is described as

$$\bar{u}(t) = C_{VR}(\rho, z)\big(\bar{r}(t) - y(t)\big). \tag{7}$$

[Step 5] When the virtual control input and the acquired control input data from Step 1 are close, the closed-loop can be regarded as close to the reference model. Therefore, the cost function can be minimized to be

$$J_{VR} = \|u(t) - \bar{u}(t)\|_2^2. \tag{8}$$

From equations (6) and (7), the above equation becomes as follows.

$$J_{VR}(\rho) = \|u(t) - C_{VR}(\rho, z)\bar{e}\|_2^2, \tag{9}$$

where

$$\bar{e}(t) = \bar{r}(t) - y(t). \tag{10}$$

[Step 6] The introduction of the prefilter L: The term in Eq. (9) has the inverse matrix of the reference model, which means that it has a nonproper characteristic. By adding a prefilter, nonproper characteristics can be avoided. Equation (11) is given by adding the prefilter to Equation (9).

$$J_{VR}(\rho) = \|u_L(t) - C_{VR}(\rho, z)\bar{e}_L(t)\|_2^2, \tag{11}$$

where

$$u_L(t) = Lu(t), \bar{e}_L(t) = L\bar{e}(t). \tag{12}$$

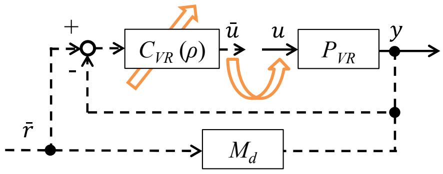

Figure 3 The VRFT concept

3. Direct tuning of the PID gain scheduler

3.1. Gain-scheduled PID control

In the gain-scheduled PID control, the closed-loop system may become unstable due to the sudden change in the gain. As a countermeasure, the velocity form of the PID controller and the described scheduling function in the polynomial are adopted.

3.1.1. Velocity form of the PID controller

The velocity form of the PID control law that is suitable for the gain-scheduled control is adopted. This has the advantage that the integral term does not need to be reset, and the control input does not rapidly change even when the gain rapidly changes [32, 33]. If the gain abruptly changes, the control input changes over time, causing disturbance to the system. Figure 4 shows a block diagram of the velocity form of the PID control. An integral element appears just before the control input, and the time change can be reduced. Furthermore, an integral element is used after the K_i element, and the K_d element is used after the difference element. However, it is not preferable when these orders are reversed, as the time change of the control input at the time of switching becomes large [34]. The velocity form of the PID control that is shown in Figure 4 is expressed by the following equation.

$$u(t) = u(t-1) + C_v(z, \rho)e(t) \tag{13}$$

with

$$\begin{gathered} C_v(z, \rho) = K(t)\psi(z), \\ K(t) = [K_p(t) \quad K_i(t) \quad K_d(t)]^T, \\ \psi(z) = [(1 - z^{-1}) \quad 1 \quad (1 - z^{-1})^2]^T, \end{gathered} \tag{14}$$

where $e(t)$ is the error that is given by $e(t) = r(t) - y(t)$, and $r(t)$ is the set-point. $K_p(t)$, $K_i(t)$, and $K_d(t)$ denote the proportional gain, integral gain, and derivative gain, respectively, and Δ represents the difference operator, which is expressed as $\Delta = 1 - z^{-1}$ using the backward operator z^{-1}, where $z^{-1}y(t) := y(t-1)$.

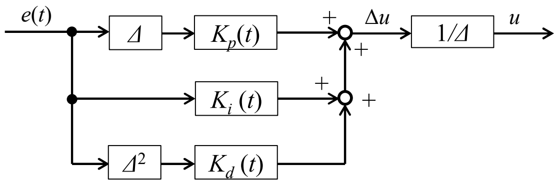

Figure 4 Block diagram of the velocity form of the PID controller

3.1.2. PID gain scheduler

In this paper, the gain scheduler in Equation (2) uses a polynomial. The use of the Just-In-Time method [30], database control [23], and neural network [35] is difficult to install in mass product controllers due to the limitations of the calculation cost and ROM area. The gain schedulers that use the LUTs have been used for a long time in many industries, especially in automobile control. However, the ROM capacity and tuning parameters increase. Furthermore, as a concern of the gain-scheduled control in which the gain that is designed for each operating point is directly expressed by the LUT, there is a possibility that the PID gain rapidly fluctuates and that the system becomes unstable. Therefore, in this paper, the scheduling function is represented by the quadratic polynomial shown in Equation (15). As a result, in addition to reducing the number of stored parameters, the gain continuously changes so that the sudden gain changes are less likely to occur.

$$\begin{gathered} K_j(\theta) = w_j^T \theta_{sf}\,, \\ w_j = \begin{bmatrix} w_{j0} & w_{j1} & w_{j2} & w_{j3} & w_{j4} & w_{j5} \end{bmatrix}^T, \\ j \in \{p, i, d\}, \\ \theta_{sf}(\theta) = [1 \quad \theta_1 \quad \theta_2 \quad \theta_1\theta_2 \quad \theta_1^2 \quad \theta_2^2]^T, \\ \theta = [\theta_1 \quad \theta_2]^T, \end{gathered} \tag{15}$$

where $K_j(\theta)$ is the PID gain scheduler ($j = p, i, d$), which is the scheduling function. θ_l is the scheduling parameter (l = 1, 2), and θ_{sf} is the function vector, which is the base function that is composed of the scheduling parameters. In addition, w_j is the weight coefficients vector, which is the regression coefficient vector of the PID gains, which are the tuning parameters. As scheduling parameters, the signals of the external environment, such as the temperature, or the states, such as the position and speed of the controlled object, are generally used. In Equation (15), the number of scheduling parameters is two, and the scheduling function is a quadratic polynomial, but it is not limited to this order.

3.2. Derivation of the cost function

Here, we derive a cost function that finds the optimum values of the weight coefficients of the gain scheduling function. From the cost function of the VRFT (see Equations (11, 12)) and Equations (13–15), which are related to the gain-scheduled PID control, the weight coefficient of the gain scheduler can be obtained. The cost function is

$$J(w) = \|d(t) - w^T \xi(t)\|_2^2, \tag{16}$$

$$\begin{aligned} d(t) &= L\Delta u(t), \\ \xi(t) &= X(M_d^{-1}(z) - I)Ly(t), \end{aligned} \tag{17}$$

with

$$w = \begin{bmatrix} w_p & w_i & w_d \end{bmatrix}^{\mathrm{T}}, \tag{18}$$

$$X = \begin{bmatrix} \theta_{sf}^T \psi_1\,(z) & \theta_{sf}^T \psi_2\,(z) & \theta_{sf}^T \psi_3\,(z) \end{bmatrix}^{\mathrm{T}}, \tag{19}$$

where ψ_i is the i−th element of the vector ψ, as shown in Equation (14). Since the cost function is linear with respect to the weight coefficients vector w, the optimal solution w^* can be obtained by the following equation using the least-squares (LS) method.

$$w^* = (Z^T Z)^{-1} Z^T D, \tag{20}$$

where

$$Z = [\xi(1) \quad \xi(2) \quad \cdots \quad \xi(N)]^{\mathrm{T}}, \tag{21}$$

$$D = [d(1) \quad d(2) \quad \cdots \quad d(N)]^{\mathrm{T}}. \tag{22}$$

3.3. Automatic tuning by LASSO

LASSO is a method for extracting essential low-dimensional information from high-dimensional information with high accuracy. By introducing the L_1 regularization term, the weight coefficients with less influence are set to zero. This property is called sparseness. In the field of machine learning, it is used to reduce the prediction error of the model by preventing overfitting. In this paper, LASSO is applied to find the optimal solution of Equation (16). As a result, it can be expected to suppress the overfitting of the weight coefficients and to obtain a controller with high sparse.

The cost function obtained by adding the L_1 regularization term to Equation (16) is

$$J(w) = \|d(t) - w^T \xi(t)\|_2^2 + \lambda \|w\|_1, \tag{23}$$

where λ is a positive constant and a parameter that tunes the relative strength between the regularization term and the sum of the squared error terms. The sparsity can be tuned by changing the value of λ. In this paper, cross-validation is used to automatically determine the optimal regularization parameter λ from the multiple λ, which is set by the designer. The cross-validation can be explained using Figure 5.

[Step 1] Divide the data into k (=5) blocks. This is called a fold.

[Step 2] Fold 1 and the others (folds 2–5) are used as the test and training sets, respectively. Then, the learning and evaluation of the model are performed, and the cost function value J_1 is obtained.

[Step 3] Fold 2 and the others (folds 1, 3–5) are used as the test and training sets, respectively. Then, the learning and evaluation of the model are performed, and the cost function value J_2 is obtained.

[Step 4] This process is repeated until fold 5 becomes the test set, and the cost functions J_1–J_5 can finally be obtained. Also, the cost value J, which is the average value of J_1–J_5, can be calculated.

By performing cross-validation for multiple λ values, which are set by the designer, and by obtaining the cost function value J for each λ, the optimum λ and the optimum parameters with the smallest cost functions are obtained. Python is used as the programming language, and cross-validation is performed by LASSO CV, which is included in the package scikit-learn to determine λ and to find the optimal solution.

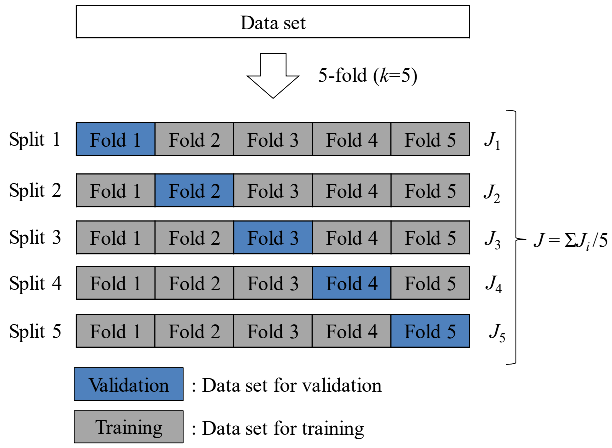

Figure 5 k-fold cross-validation (k is five in the figure.)

3.4. Algorithm

Using the VRFT, the algorithm for the direct tuning method of the weight coefficients of the PID gain scheduling function is shown below.

[Step 1] Measure the input and output data $u(t)$, $y(t)$, $t = 1, ..., N$ in the test. The data set is Z and D described in Equations (21) and (22), respectively.

[Step 2] Set the reference model.

[Step 3] Decide the scheduling parameter candidates and the scheduling function for each PID gain.

[Step 4] Design the prefilter for the VRFT.

[Step 5] Obtain the weight coefficients, which are the scheduling parameters, of the scheduling function that minimizes the cost function by LASSO.

Remark 1. In Step 4, by applying a strict prefilter in the study [27–29], the original cost function and the VRFT cost function can be matched. However, additional experiments are required. In this paper, we consider the practical application and use the prefilter of the following filter [18]:

$$L(z) = M_d(z). \tag{24}$$

Remark 2. This algorithm is based on VRFT, where an open-loop test has been adopted, as in many studies [9, 17–19, 27–29]. This algorithm faces the same theoretical problem as the general system identification and the previous studies [27–29] when using closed-loop test data. For practice use, we could obtain controller parameters realizing model-matching if the noise is of an acceptable magnitude. The literature [17] shows that optimized parameters can be obtained using closed-loop data for an industrial system with the standard VRFT. If the observed noise is large, a filtering process is required similar to the common machine learning process. In addition, in FRIT, which has a lot of achievements using closed-loop test data, a filtering process is performed for noisy data

[36]. In Section 4, we will show the results when using the initial data of open-loop and closed-loop tests. The results show that good performance was achieved in both cases.

Remark 3. We summarize the feature of this method. The advantages of the proposed method are described as follows: (i) The gain-scheduled controller parameters can be obtained without the controlled object model. (ii) ROM area and calculation cost can be reduced using LASSO. In addition, LASSO prevents overlearning. (iii) Many engineers can easily understand the controller structure, as industrial engineers are familiar with gain-scheduled control and PID control. On the other hand, the disadvantage is that stabilization is not completely guaranteed; however, this is a common issue with direct tuning methods based on data-driven controllers [2, 9, 11, 17, 18, 23–26, 34]. This will be one of our future works.

4. Simulation validation

The controlled objects are two nonlinear systems. The first one is a nonlinear dynamic system described in the LPV system, and it is a spring-mass system that is often used in industrial systems, where each parameter is time-varying. The second system is the Hammerstein model [37], in which a linear dynamic system is connected in series to the output of the static nonlinear function, and it is widely used as a model for describing the nonlinear system [38]. This model is also used as a verification model for data-driven control [23–25, 39]. We use open-loop and closed-loop data to validate the effectiveness of the proposed method for these systems.

4.1. Application to LPV systems

It is applied to a controlled object whose system parameters change according to the state of the controlled object. In other words, the target is a nonlinear dynamic system.

4.1.1. System formulation

The controlled object, reference model, and controller used in the simulation verification are formulated. The controlled object is a spring-mass system with time-varying parameters, as shown in Figure 6. Here, m, c, k, and y denote the mass, viscosity coefficient, spring stiffness, and system response (position), respectively. The mass, spring stiffness, and viscosity coefficient change depending on the system response and the controlled object is a system in which the following equation of motion is discretized.

$$m(y,t)\frac{d^2y(t)}{dt^2}+c(y,t)\frac{dy(t)}{dt}+k(y,t)=u(t)+v(t), \tag{25}$$

where

$$\begin{aligned} m(y,t) &= 1+0.2y(t), \\ k(y,t) &= 5+2y+y^2(t), \\ c(y,t) &= 2+0.5y(t). \end{aligned} \tag{26}$$

v is the white noise with the variance 1×10^{-4}, and the set-point at each time is given as

$$r(t) = \begin{cases} 0.75 \ (0 < t \leq 10) \\ 2.0 \ (10 < t \leq 25) \\ 1.25 \ (25 < t \leq 40) \\ 0.5 \ (40 < t \leq 50) \end{cases}. \tag{27}$$

The reference model is a first-order system with a time constant of 1 s as

$$\begin{aligned} M_d(z) &= z\big(M_d(s)\big), \\ M_d(s) &= \frac{1}{s+1}, \end{aligned} \tag{28}$$

where $z(\cdot)$ represents the discretization, s is the Laplace operator, and the sampling period of the controller is 10 ms. The gain scheduler uses Equation (15), and the scheduling parameters are the position and its derivative as

$$\theta_1(t) = y(t), \theta_2(t) = y(t)(1 - z^{-1}). \tag{29}$$

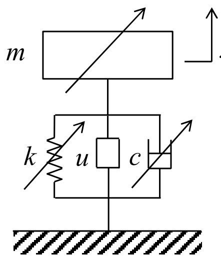

Figure 6 Controlled object with time-varying parameters

4.1.2. Result

Figure 7 shows the given input/output data in the open-loop test. The input signal is given to the staircase signal to obtain the input/output data that represents the characteristics around the equilibrium points, and the step width is 3.0. From the figure, it can be seen that the system becomes difficult to move with the increase in the position. Figure 8 shows the time series data when the gain scheduling parameters were obtained from the input/output data in the open-loop test. For comparison, the data when using the fixed PID gain (VRFT), the VRFT by GS (using the least-squares method (GS-VRFT-LS)), and the VRFT by GS (using LASSO regression (GS-VRFT- LASSO)) are shown. The fixed PID gains obtained by the VRFT are $K_p = -0.0340$, $K_i = 0.0605$, and $K_d = 0.0070$, respectively. From the top of the figure, the output, input, proportional gain, integral gain, and derivative gain are shown. Table 1 shows the MSE (mean squared error) of the tracking error performance, which is $\mathrm{MSE} = \frac{1}{N}\sum_{t=1}^{N}\big(y(t) - M_d(z)r(t)\big)^2$, of VRFT, GS-VRFT-LS, and GS-VRFT-LASSO As shown in the figure and table, the GS-VRFT-LS has a better tracking performance than the standard VRFT. This is because the PID gain changes appropriately according to the state of the controlled object in the GS-VRFT-LS. Next, by comparing GS-VRFT-LS and GS-VRFT-LASSO, it can be seen that they have the same tracking characteristics. Table 2 shows the operation counts and weight coefficients obtained by the GS-VRFT-LS and GS-VRFT-LASSO for calculating PID gains from the gain scheduler in Equation (15). The total counts of multiplication and addition in the scheduler obtained using LASSO are less than those obtained using the least-squares method. In the scheduler obtained by LASSO, the 18 weight coefficients that make up the PID gain scheduler are reduced to 9, and the same performance is obtained when using 18 weight coefficients. The weight coefficients w_2, w_3, and w_5 of the PID gain scheduler became zeros. In other words, the gain-scheduled PID controller with high sparsity is obtained. The optimum value of λ for the L_1 regularization was 1×10^{-7} because of the cross-validation for multiple λ, which is described in Section 3.3.

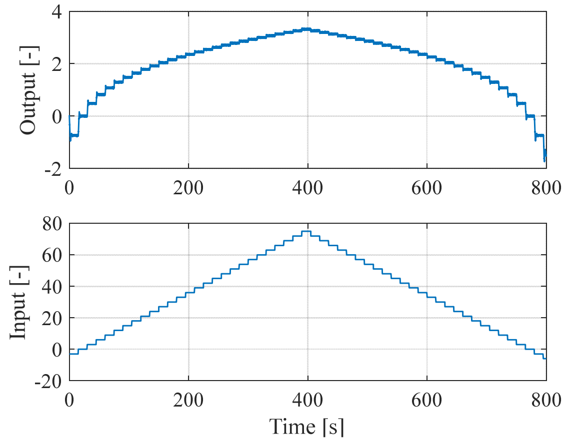

Figure 7 Time series data of the initial input and output under open-loop test

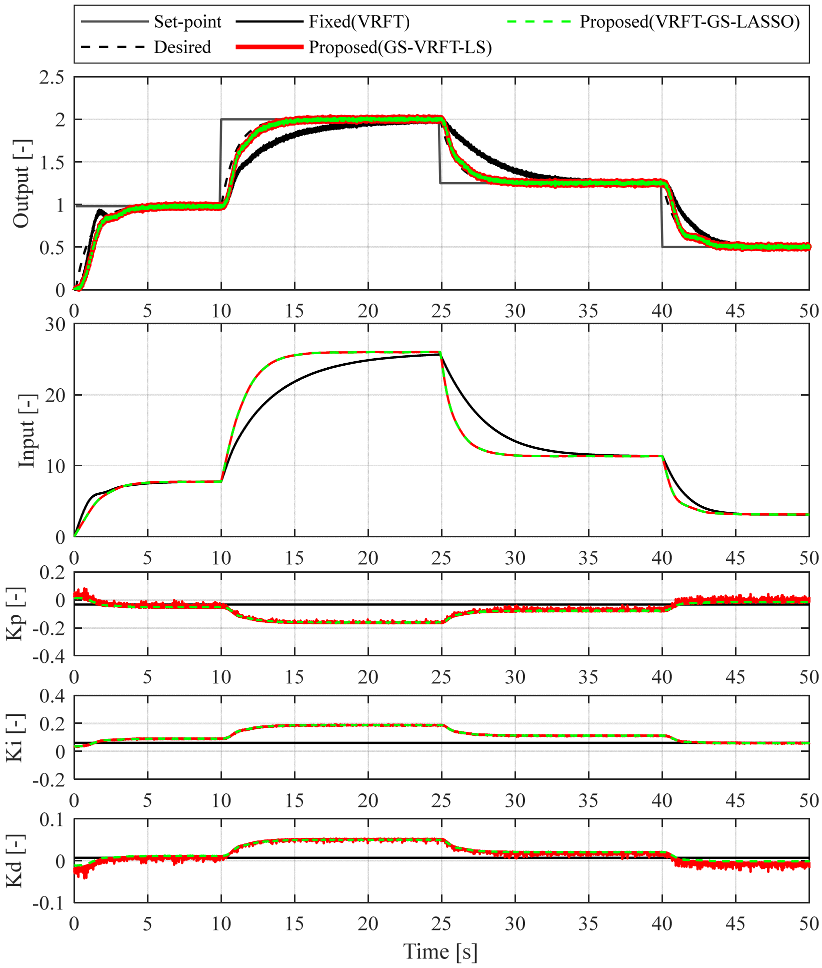

Figure 8 Time series data with the proposed method and fixed PID gain using open-loop data (The GS-VRFT-LS and GS-VRFT-LASSO are almost overlapped.)

Table 1. Results of the tracking error for the spring-mass model for open-loop test

	Fixed (VRFT)	GS-VRFT-LS	GS-VRFT-LASSO
MSE	1.918×10^{-2}	4.149×10^{-3}	4.178×10^{-3}

Table 2. Operation counts and parameters of gain scheduler for spring-mass system

	GS-VRFT-LS	GS-VRFT-LASSO	Efficiency [%]
Multiplication	33	12	36.4
Addition	15	6	40.0
Total	48	18	37.5
Weight coefficients	18	9	50.0

Figure 9 shows the given input/output data in the closed-loop test. The set-point signal is given to the staircase signal of which step width is 0.25, and the input/output data is measured. The fixed PID gains are set to K_p = 0.0, K_i = 0.3, and K_d = 0.0. Figure

10 shows the time series data obtained from the closed-loop test input/output data when gain scheduling parameters are used. For comparison, the data when using the fixed PID gain (VRFT), VRFT by the GS (using the least-squares method (GS-VRFT-LS)), and VRFT by GS (using LASSO regression (GS-VRFT- LASSO)) are shown. The fixed PID gains obtained using VRFT are $K_p = -0.0651$, $K_i = 0.0684$, and $K_d = 0.0212$. Table 3 shows the MSE of the tracking error performance of VRFT, GS-VRFT-LS, and GS-VRFT-LASSO. From the figure and table, we can see that the performance using closed-loop experiment data is good as well as the results using open-loop experiment data. In addition, the operation counts and weight coefficients obtained using the GS-VRFT-LS and GS-VRFT-LASSO are the same as those shown in Table 2. The total counts of multiplication and addition in the scheduler obtained using LASSO are less than those obtained using the least-squares method. In the scheduler obtained using LASSO, the 18 weight coefficients that make up the PID gain scheduler are reduced to 9, and the same performance is obtained when using 18 weight coefficients. The weight coefficients w_2, w_3, and w_5 of the PID gain scheduler became zeros. In other words, a gain-scheduled PID controller with high sparsity was obtained. The optimum value of λ for the L_1 regularization was 1×10^{-7}.

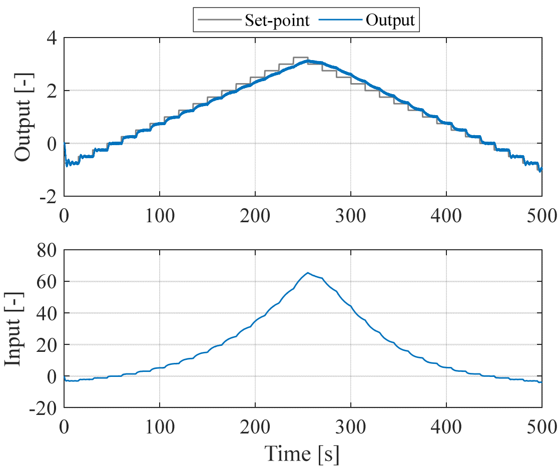

Figure 9 Time series data of the initial input and output under closed-loop test.

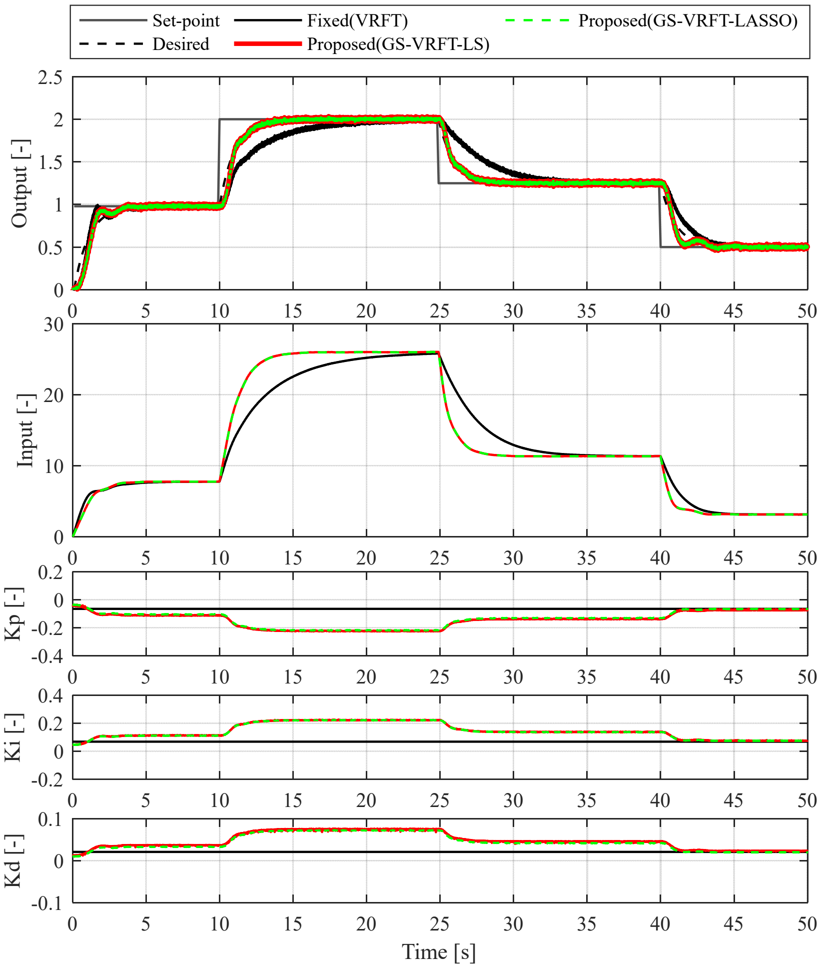

Figure 10 Time series data with the proposed method and fixed PID gain using closed-loop data (The GS-VRFT-LS and GS-VRFT-LASSO are almost overlapped.)

Table 3. Results of the tracking error for the spring-mass model for closed-loop

	Fixed (VRFT)	GS-VRFT-LS	GS-VRFT-LASSO
MSE	1.474×10^{-2}	2.846×10^{-3}	2.834×10^{-3}

4.2. Application to the Hammerstein model

The Hammerstein model, which is widely used as a model for describing nonlinear systems, is the controlled object.

4.2.1. System formulation

The system formulation in this section is the same as that in the previous literature [23–25, 37]. The sampling period of the simulation, which includes the controller, is $1s$, and the Hammerstein model is the control target, as shown in the following equation.

$$\begin{gathered} y(t) = 0.6y(t-1) - 0.1y(t-2) + 1.2x(t-1) \\ -0.1x(t-2) + v(t), \\ x(t) = 1.5u(t) - 1.5u^2(t) + 0.5u^3(t), \end{gathered} \tag{30}$$

where v is the white noise with the variance 1×10^{-3}. The set-point at each time is

$$r(t) = \begin{cases} 1.0 \ (0 < t \leq 100) \\ 3.0 \ (100 < t \leq 200) \\ 0.5 \ (200 < t \leq 300) \\ 2.0 \ (300 < t \leq 400) \end{cases}. \tag{31}$$

The reference model uses the following equation [23–25].

$$M_d(z^{-1}) = \frac{0.399z^{-1}}{1 - 0.736z^{-1} + 0.135z^{-2}} \tag{32}$$

The gain scheduler uses Eq. (15), and the scheduling parameters are the output, and its second derivative is

$$\theta_1(t) = y(t), \theta_2(t) = y(t)(1 - z^{-1})^2. \tag{33}$$

In Section 4.1, we adopted the derivative as a scheduling parameter. In Section 4.2, we adopted the second derivative, which is more susceptible to noise, as another candidate.

4.2.2. Result

Figure 11 shows the given input/output data in the open-loop test. A chirp sin signal (frequency 0 to 1 Hz, amplitude 1.75, offset 1) is applied to the input, and the input/output data is measured. Figure 12 shows the time series data after the gain scheduling parameters are obtained from the measured input/output data. For comparison, this figure shows the time series data when using the CHR method, the standard VRFT (fixed PID gain), the VRFT by GS (applying the least-squares method (GS-VRFT-LS)), and the VRFT by GS (applying LASSO (GS-VRFT-LASSO)). The PID gains by the CHR method are $K_p = 0.059$, $K_i = 0.058$, and $K_d = 0.0038$, which were obtained from previous studies [23–25]. The fixed PID gains obtained by VRFT are $K_p = 0.0655$, $K_i = 0.1744$, and $K_d = 0.0166$. From the top of the figure, output, input, proportional gain, integral gain, and derivative gain are shown. Table 4 shows the MSE of the tracking error performance of CHR, VRFT, GS-VRFT-LS, and GS-VRFT-LASSO. In addition, as shown in the figure and the table, the response was very slow in the classical CHR method. By comparing the standard VRFT and GS-VRFT-LS, we can confirm that the GS-VRFT-LS has a PID gain that changes according to the state of the controlled object and that follows the target response. Also, by comparing the GS-VRFT-LS and GS-VRFT-LASSO, it can be seen that they have the almost same followability. Table 5 shows operation counts and weight coefficients obtained using the GS-VRFT-LS and GS-VRFT-LASSO for calculating PID gains from the gain scheduler shown in Equation (15). The total counts of multiplication and addition in the scheduler obtained using LASSO are less than those obtained using the least-squares method. In LASSO, two weight coefficients became zero. In other words, the gain scheduling was performed with 18 weight coefficients for the LS and 16 weight coefficients for LASSO. The optimum value of λ for the L_1 regularization was 0.002 because of the cross-validation for multiple λ, which is described in Section 3.3.

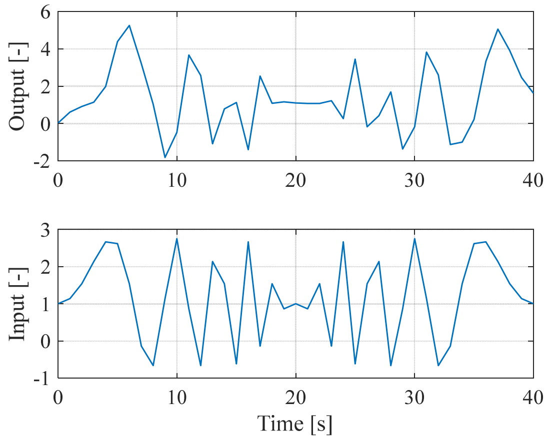

Figure 11 Time series data of the initial input and output under open-loop test

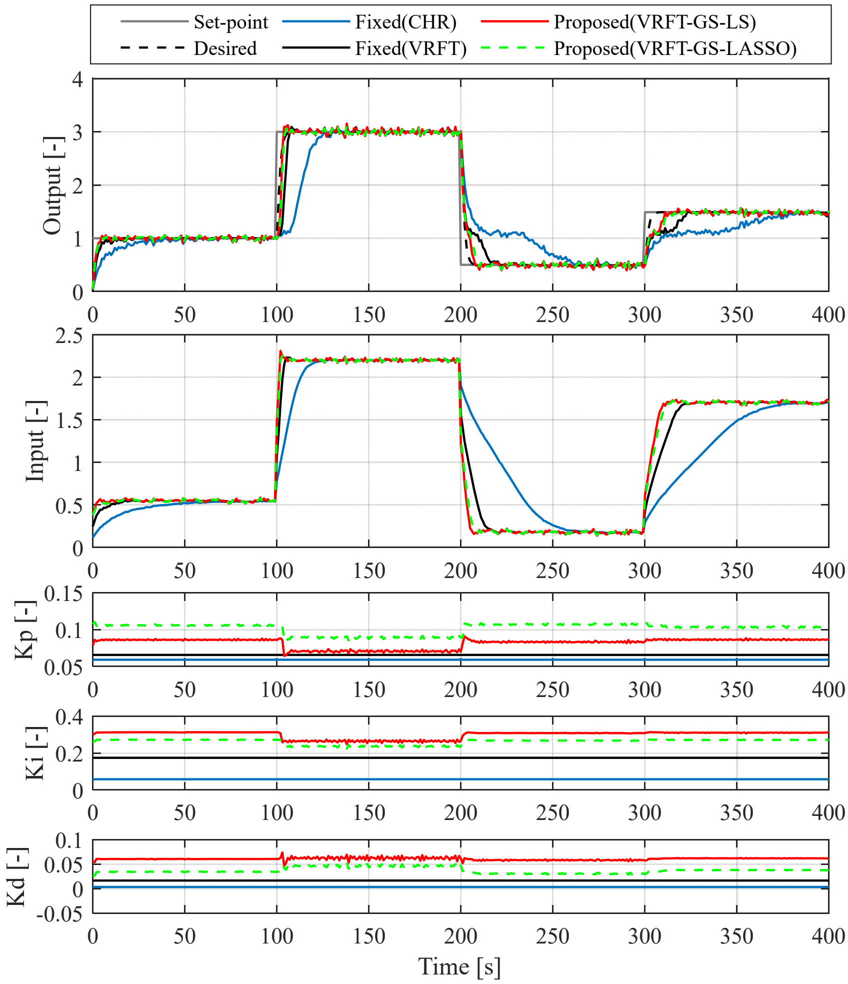

Figure 12 Time series data with the proposed method and fixed PID gain using open-loop data (GS-VRFT-LS and GS-VRFT-LASSO are overlapped.)

Table 4. Results of the tracking error for the Hammerstein model for open-loop test

	Fixed (CHR)	Fixed (VRFT)	GS-VRFT-LS	GS-VRFT-LASSO
MSE	1.548×10^{-1}	2.830×10^{-2}	8.121×10^{-3}	1.053×10^{-2}

Table 5. Operation counts and parameters of gain scheduler for Hammerstein using open-loop data

	GS-VRFT-LS	GS-VRFT-LASSO	Efficiency [%]
Multiplication	33	30	90.9
Addition	15	13	86.7
Total	48	43	89.6
Parameters	18	16	88.9

Figure 13 shows the given input/output data in the closed-loop test. The set-point signal is given to the random signal of which range is from −1 to 5, and the input/output data are measured. The fixed PID gains are set to $K_p = 0.059$, $K_i = 0.058$, and $K_d = 0.0038$, which were obtained using the CHR method in previous studies [23–25]. Figure 14 shows the time series data when gain scheduling parameters were obtained from the input/output data in the closed-loop test. For comparison, the data obtained when using the fixed PID gain (VRFT), the VRFT by GS (using the least-squares method (GS-VRFT-LS)), and the VRFT by GS (using LASSO regression (GS-VRFT- LASSO)) are shown. The fixed PID gains obtained using VRFT are $K_p = 0.0206$, $K_i = 0.1246$, and $K_d = 0.0229$. Table 6 shows the MSE of tracking error performance of CHR, VRFT, GS-VRFT-LS, and GS-VRFT-LASSO. Table 7 shows the operation counts and weight coefficients obtained using GS-VRFT-LS and GS-VRFT-LASSO for calculating PID gains from the gain scheduler shown in Equation (15). The total counts of multiplication and addition in the scheduler obtained by LASSO are less than those obtained using the least-squares method. In LASSO, one weight coefficient became zero. In other words, the gain scheduling was performed with 18 weight coefficients for the LS and 17 weight coefficients for LASSO. The optimum value of λ for the L_1 regularization was 1×10^{-5} because of the cross-validation for multiple λ, which is described in Section 3.3.

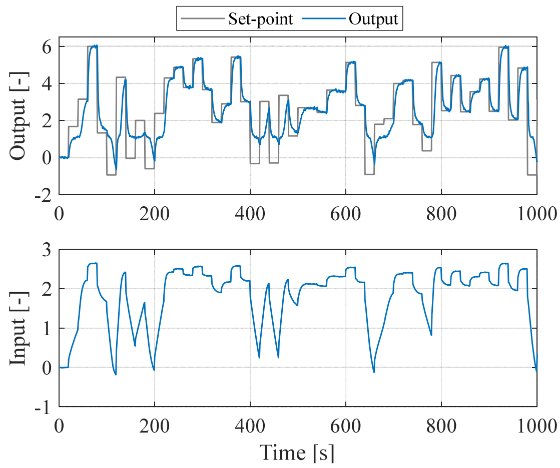

Figure 13 Time series data of the initial input and output under the closed-loop

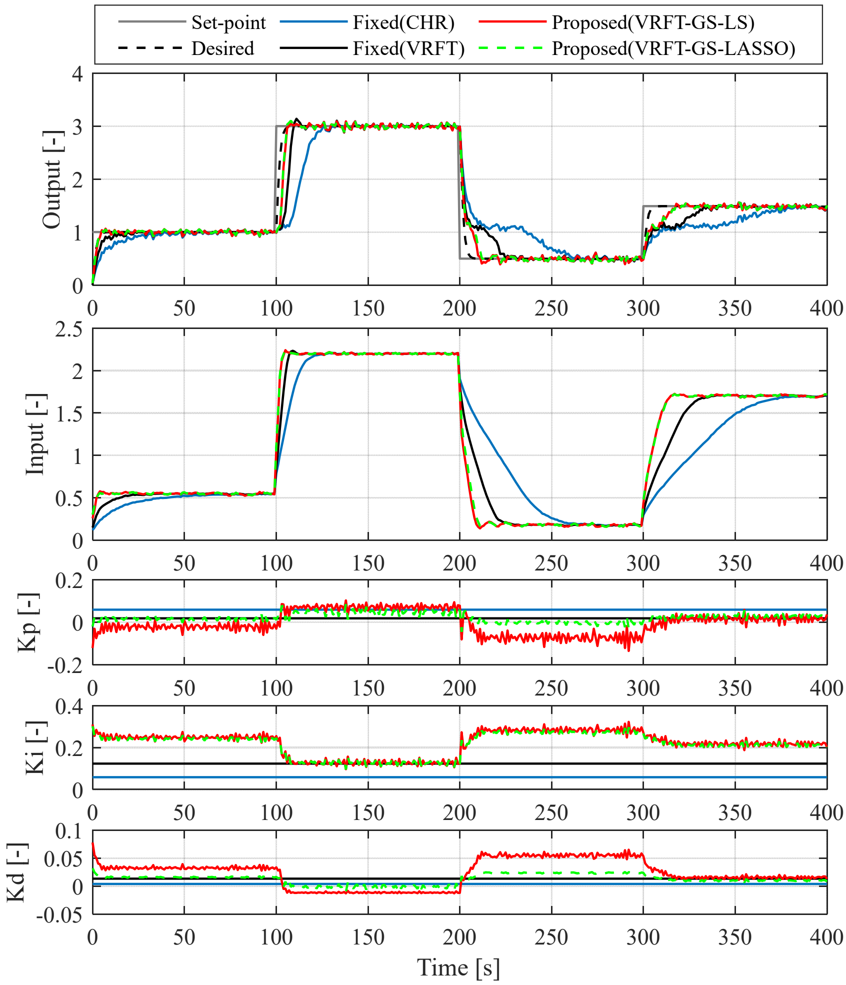

Figure 14 Time series data with the proposed method and fixed PID gain using closed-loop data (GS-VRFT-LS and GS-VRFT-LASSO are overlapped.)

Table 6. Results of the tracking error for Hammerstein model for closed-loop test

	Fixed (CHR)	Fixed (VRFT)	GS-VRFT-LS	GS-VRFT-LASSO
MSE	1.548×10^{-1}	5.562×10^{-2}	1.884×10^{-2}	1.908×10^{-2}

Table 7. Operation counts and parameters of gain scheduler for Hammerstein model using closed-loop data

	GS-VRFT-LS	GS-VRFT-LASSO	Efficiency [%]
Multiplication	33	31	93.9
Addition	15	14	93.3
Total	48	45	93.8
Parameters	18	17	94.4

4.3. Discussion

When applied to System 2, the number of reductions of the weight coefficients by LASSO was low. However, System 1 was an LPV system whose characteristics changed because of the position. Therefore, it is considered that the velocity, which is the first derivative of the position used as the scheduling parameter, has little effect on the system characteristic fluctuation and that the number of weight coefficients is reduced. As a result, the calculation cost of the controller and the ROM area can be reduced, which is a significant result for the implementation in the mass product controllers. In System 1, the closed-loop system became unstable when the weight coefficients were obtained by the LS method under the simulation condition that scheduling parameters are positions and acceleration without the white noise. However, it was stable when LASSO was used. This is because overlearning occurred due to the unnecessary weight coefficients related to the acceleration when the LS method was used; however, in LASSO, the weight coefficients for acceleration were zero, implying that overlearning was suppressed. Regarding the use of open-loop and closed-loop test data, we showed the tracking performance of the proposed method is significantly better than those of the conventional fixed gain tuning methods in both cases. However, in system 2, the trajectory of PID gains between open-loop and closed-loop is different. This is because initial input/output data are different between open-loop and closed-loop tests. We consider that multiple candidates exist for the controlled variable to be close to the reference. In Section 4, we used the position as one of the scheduler parameters, and another was velocity and acceleration for system 1 and system 2, respectively. Contrary to the above cases, it was confirmed that the same performance can be obtained when acceleration and velocity are used as scheduler parameters for system 1 and system 2, respectively.

5. Conclusion

In this paper, we proposed a design method of a data-driven gain-scheduled PID controller that considers sparseness without system identification for two types of nonlinear systems. In this method, to reduce the tuning parameters, a polynomial was used as the scheduling function, and the weight coefficients of the scheduling function, which are the tuning parameters, were obtained based on the data-driven control. By applying the VRFT, a gain-scheduled PID controller could be directly designed from a set of input/output data without system identification. Furthermore, in the optimization, LASSO was used to further reduce the controller parameters. The effectiveness of this method was examined by simulation for two types of nonlinear systems. As a result, it was revealed that a controller with high sparse can be obtained without knowing the characteristics of the controlled object for a large number of control parameters of the gain scheduler. In summary, it was possible to realize gain-scheduled PID control with a low computational calculation cost and ROM area and without trial and error parameter tuning. We consider that the proposed control law and tuning method can easily be accepted in many industries. In the future, we plan to apply it to industrial systems with strong nonlinearity, such as internal combustion engines and automatic transmissions.

6. References

1. K. J. Åström, T. Hägglund, *The future of PID control*, Control Engineering Practice, **9**(2001), 1163-1175.
2. Y. Liao, K. Koiwai, T. Yamamoto, *Design and implementation of a hierarchical‐clustering CMAC PID Controller*, Asian Journal of Control, **21**(2019), 1077-1087.
3. F. Wu, A. Packard, G. Balas, *Systematic gain‐scheduling control design: A missile autopilot example*, Asian Journal of Control, **4**(2002), Issue3, 341-347.

4. V. Veselý, A. Ilka, *Gain-scheduled PID controller design*, Journal of Process Control, **23**(2013), 1141-1148.
5. W. J. Rugh, J. S. Shamma, *Research on gain scheduling*, Automatica, **36**(2000), 1401-1425.
6. E. B. Ondes, I. Bayezit, I. Poergye, A. Hafsi, *Model-based 2-D look-up table calibration tool development*, 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, Australia.
7. 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, **57**(2013), no. 7, 1979-1989.
8. X. Bu, Y. Qiao, Z. Hou, J. Yang, *Model free adaptive control for a class of nonlinear systems using quantized information*, Asian Journal of Control, **20**(2018), 962-968.
9. M. C. Campi, A. Lecchini, S. M. Savaresi, *Virtual reference feedback tuning: A direct method for the design of feedback controllers*, Automatica, **38**(2002), 1337-1346.
10. S. Soma, O. Kaneko, *A new method of controller parameter tuning based on input-output data–Fictitious Reference Iterative Tuning (FRIT)*, IFAC Proceedings Volumes, **37**(2004), 789-794.
11. O. Kaneko, *Data-driven controller tuning: frit approach*, IFAC Proceedings, **46** (2013), 326-336.
12. S. Ishizuka, I. Kajiwara, *Online adaptive PID control for MIMO systems using simultaneous perturbation stochastic approximation*, Journal of Advanced Mechanical Design, Systems, and Manufacturing, **9** (2015), Issue 2, 1-16.
13. R. A. Fahmy, R. I. Badr, F. A. Rahman, *Adaptive PID controller using RLS for SISO stable and unstable systems*, Advances in Power Electronics, **Article ID 507142**(2014), 1-5.
14. R. A. Fahmy, R. I. Badr, F. A. Rahman, *Alternative approach to use RLS algorithm in multivariable online adaptive PID controllers for MIMO systems*, IETE Journal of Research, **64**(2018), 27-35.
15. Y. Wakasa, K. Tanaka, Y. Nishimura*, online controller tuning via FRIT and recursive least-squares*, IFAC Conference on Advances in PID Control PID'12 Brescia (Italy) 2012.
16. Y. Wakasa, S. Adachi, K. Tanaka, S. Nakashima, *FRIT and RLS-based online controller tuning and its experimental validation*, 9th Asian Control Conference (ASCC), 2013.
17. M. Nakamoto, *An Application of the virtual reference feedback tuning method for multivariable process control*, Transactions of the Society of Instrument and Control Engineers, **17**(2005), 528-536.
18. S. Formentin, S.M. Savaresi, L. D. Re, *Non-iterative direct data-driven controller tuning for multivariable systems theory and application*, IET Control Theory and Applications, **6**(2011), 1250-257.
19. J. D. Rojas, X. Flores-Alsina, U. Jeppsson, R. Vilanova, *Application of multivariate virtual reference feedback tuning for wastewater treatment plant control*, Control Engineering Practice, **20**(2012), 499-510.
20. S. Ishizuka, I. Kajiwara, J. Sato, Y. Hanamura, S. Hanawa, *Model-free adaptive control scheme for EGR/VNT control of a diesel engine using the simultaneous perturbation stochastic approximation*, Transactions of the Institute of Measurement and Control, **39**(2017), 114-128.

21. I. Kajiwara, K. Furuya, S. Ishizuka, *Experimental verification of real-time tuning method of a model-based controller by perturbations to its poles*, Mechanical Systems and Signal Processing, **107**(2018), 396-408.
22. S. Yahagi, I. Kajiwara, 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, **234**(2020), 2279-2290.
23. T. Yamamoto, K. Takao, T. Yamada, Design of a Data-Driven PID Controller, IEEE Transactions on Control Systems Technology, **17** (2009), Issue 1, 29–39.
24. S. Wakitani, Y. Ohnishi, T. Yamamoto, *Design of FRIT-based nonlinear PID control systems*, Journal of The Society of Instrument and Control Engineers, **52**(2013), 885-891. [in Japanese]
25. S. Wakitani, K. Nishida, M. Nakamoto, T. Yamamoto, *Design of a data-driven PID controller using operating data*, IFAC Proceedings Volumes, **46**(2013), 587-592.
26. F. Tsukui, S. Masuda, *Data-driven control parameter tuning using feedback linearization*, IEEJ Transactions on Electronics, Information and Systems, **137**(2017), 891-897. [in Japanese]
27. S. Formentin, G. Panzani, S. M. Savaresi, *VRFT for LPV systems: Theory and braking control application*, Robust Control and LPV Approaches, (2013), 289-309.
28. S. Formentin, S. M. Savaresi, *Virtual Reference Feedback Tuning for linear parameter-varying systems*, Proceedings of the 18th World Congress, Milano (Italy), (2011), 10219-10224.
29. S. Formentin, D. Piga, R. Toth, Savaresi SM, *Direct learning of LPV controllers from data*, Automatica, **95**(2016), 98-110.
30. A. Stenman, F. Gustafsson, L. Ljung, *Just in time models for dynamical systems*, Proceedings of IEEE International Conference on Decision and Control, (1996), 1115-1120.
31. Z. Hou, S. Jin, *Data-driven model-free adaptive control for a class of MIMO nonlinear discrete- time systems*, IEEE Transactions on neural networks, **22**(2011), 2173-2188.
32. N. Suda, *PID Control*. Asakura Shoten, 1992. [in Japanese]
33. K. J. Åström, T. Hägglund, *PID controllers: Theory, Design and Tuning, 2nd edition*, ISA, 1995.
34. M. Saeki, K. Ogawa, *Gain scheduled PID controller design by data-driven loop-shaping method*, IEEJ Transactions on Electronics, Information and Systems, **131**(2011), 758-763. [in Japanese]
35. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, 2006.
36. M. Ogawa, M. Kano, *Development of a Two-stages E-FRIT Tuning Method for Generalized Two-degrees of Freedom PID Control System*, Transactions of the Society of Instrument and Control Engineers, **11**(2016), 631-638. [in Japanese]
37. Z. Lang, *On identification of the controlled plants described by the Hammerstein model*, IEEE Trans. Automatic Control, **39**(1994), 569-573.

38. H.Tanaka, S.Hazue, *Identification of a Hammerstein Model equipped with a static nonlinear odd function, transactions of the institute of systems*, Control and Information Engineers, **20**(2007), 430-438.
39. K. Takao, T. Yamamoto, T. Hinamoto*, Design of a memory‑based prefilter supplementing a robust PID control system*, Asian Journal of Control, **10**(2008), 301-313.