INVITED PAPER

Kernel-Based Hamilton-Jacobi Equations for Data-Driven Optimal Control: The General Case

Yuji ITO[†a)], *Nonmember and* **Kenji FUJIMOTO**[††b)], *Member*

SUMMARY Recently, control theory using machine learning, which is useful for the control of unknown systems, has attracted significant attention. This study focuses on such a topic with optimal control problems for unknown nonlinear systems. Because optimal controllers are designed based on mathematical models of the systems, it is challenging to obtain models with insufficient knowledge of the systems. Kernel functions are promising for developing data-driven models with limited knowledge. However, the complex forms of such kernel-based models make it difficult to design the optimal controllers. The design corresponds to solving Hamilton-Jacobi (HJ) equations because their solutions provide optimal controllers. Therefore, the aim of this study is to derive certain kernel-based models for which the HJ equations are solved in an exact sense, which is an extended version of the authors' former work. The HJ equations are decomposed into tractable algebraic matrix equations and nonlinear functions. Solving the matrix equations enables us to obtain the optimal controllers of the model. A numerical simulation demonstrates that kernel-based models and controllers are successfully developed.

key words: *optimal control, machine learning, kernel functions, Hamilton-Jacobi equations*

1. Introduction

Important topics in the field of engineering include the control of various unknown nonlinear systems, such as autonomous vehicles interacting with human drivers [1] and batteries in electric vehicles [2]. Model-based approaches are widely used for the control of the unknown systems. In the model-based framework, after mathematical models of the systems are obtained using provided data sets, we design appropriate controllers based on the models. Various types of system models have been proposed, because the identification accuracy of the models affects the resultant control performance. Promising models involve data-driven kernel-based functions, such as Gaussian processes (GPs) [3]–[5], GP-based state-dependent coefficient models [6], and kernel ridge regression (KRR) models [7]. The GPs have been widely utilized in control applications [8]–[12]. The kernel-based functions are established without requiring sufficient knowledge regarding the systems. This advantage motivated us to focus on kernel-based functions in model-based control for unknown nonlinear systems.

Optimal control problems for kernel-based models is difficult to solve whereas the models can describe various nonlinear dynamics. Because the control design is often reduced to solving nonlinear equations involving the models, it is difficult to solve the equations. Various approaches have relaxed the control problems into solvable forms. Model predictive control has been employed to minimize predefined performance indices approximately [13]–[18]. Other relaxations are realized by differential dynamic programming [19], iterative linear quadratic regulators [20], and a gradient-based method [10], [21], [22]. Although these approaches have tackled the control problems, some or all of the following drawbacks occur because of the difficulties in the control problems. Relaxing the problems does not ensure the optimality of the designed controllers. High computational costs are required to approximately solve the problem. When the controllers are employed, stability of the system models is not guaranteed.

Therefore, we proposed a controller design method for certain kernel-based models [23] to address the aforementioned drawbacks. The main contribution of our previous work [23] was deriving kernel-based models for which optimal control problems are easily solved under several assumptions. After executing the system identification, an analytical method provides the corresponding optimal controller without approximations and large computations. Applying the obtained controller guarantees stability of the system model under the assumptions. The method is data-driven in the sense that the controller is designed using such a data-driven model.

However, our previous work [23] has a limitation regarding target systems, which restricts the range of applications. The input vector fields of the systems must be constant; thus, the effects of the control inputs to the systems are constant. Such an example is illustrated in Fig. 1. This study extends the previous work to overcome this limitation. We decompose the input vector fields into scalar nonlinear functions and constant matrices. Such a decomposition allows us to use the previous method for nonlinear input vector fields.

The remainder of this paper is organized as follows. Mathematical notation is introduced in Sect. 2. Our objective and relevant topics are described in Sect. 3. The previous work [23] to address the objective is reviewed in Sect. 4. Section 5 extends the previous work. Section 6 presents a numerical simulation to evaluate the proposed method. Finally, we conclude this study in Sect. 7.

Manuscript received March 22, 2021.
Manuscript revised June 8, 2021.
Manuscript publicized July 12, 2021.

[†] The author is with TOYOTA CENTRAL R&D LABS., INC., Nagakute-shi, 480-1192 Japan.

[††] The author is with the Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University, Kyoto-shi, 615-8540 Japan.

a) E-mail: ito-yuji@mosk.tytlabs.co.jp (Corresponding author)
b) E-mail: k.fujimoto@ieee.org

DOI: 10.1587/transfun.2021EAI0002

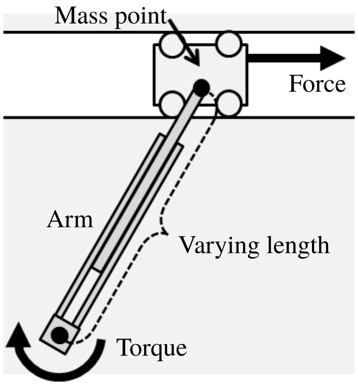


Fig. 1 Toy examples of constant and nonlinear input vector fields. The state consists of the horizontal position and velocity of the mass point. The input consists of the force and torque. Applying the input moves the mass point in the horizontal direction. The force affects the horizontal acceleration at a constant ratio, which indicates a constant input vector field. Meanwhile, the torque affects the acceleration depending on the angle and varying length of the arm, implying a nonlinear vector field.

2. Notation

This study uses the notations listed below.

- $[\boldsymbol{v}]_i$: the i-th component of $\boldsymbol{v} \in \mathbb{R}^n$
- $[\boldsymbol{A}]_{i,j}$: the i-th row j-th column entry of $\boldsymbol{A} \in \mathbb{R}^{n\times m}$
- $\mathrm{vec}(\boldsymbol{A}) := [[\boldsymbol{A}]_{1,1}, \ldots, [\boldsymbol{A}]_{n,1}, [\boldsymbol{A}]_{1,2}, \ldots, [\boldsymbol{A}]_{n,2}, \ldots, [\boldsymbol{A}]_{1,m}, \ldots, [\boldsymbol{A}]_{n,m}]^\top \in \mathbb{R}^{nm}$ for $\boldsymbol{A} \in \mathbb{R}^{n\times m}$
- $\boldsymbol{A}_\mathrm{a} \otimes \boldsymbol{A}_\mathrm{b} \in \mathbb{R}^{n_\mathrm{a} n_\mathrm{b} \times m_\mathrm{a} m_\mathrm{b}}$: the Kronecker product of $\boldsymbol{A}_\mathrm{a} \in \mathbb{R}^{n_\mathrm{a}\times m_\mathrm{a}}$ and $\boldsymbol{A}_\mathrm{b} \in \mathbb{R}^{n_\mathrm{b}\times m_\mathrm{b}}$ as follows:

$$\boldsymbol{A}_\mathrm{a} \otimes \boldsymbol{A}_\mathrm{b} = \begin{bmatrix} [\boldsymbol{A}_\mathrm{a}]_{1,1}\boldsymbol{A}_\mathrm{b} & \ldots & [\boldsymbol{A}_\mathrm{a}]_{1,m_\mathrm{a}}\boldsymbol{A}_\mathrm{b} \\ \vdots & \ddots & \vdots \\ [\boldsymbol{A}_\mathrm{a}]_{n_\mathrm{a},1}\boldsymbol{A}_\mathrm{b} & \ldots & [\boldsymbol{A}_\mathrm{a}]_{n_\mathrm{a},m_\mathrm{a}}\boldsymbol{A}_\mathrm{b} \end{bmatrix}$$

- $\partial_{\boldsymbol{v}} \boldsymbol{g}(\boldsymbol{v}) \in \mathbb{R}^{n\times m}$: the partial derivative of $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{1\times m}$ with respect to $\boldsymbol{v} \in \mathbb{R}^n$
- $\boldsymbol{I}_n$: the $n \times n$ identity matrix

3. Objective

This study focuses on a data-driven control problem for unknown systems. Section 3.1 states the goal of this study associated with a target system. Sections 3.2 and 3.3 introduce the relevant topics: kernel-based system identification and nonlinear optimal control, respectively.

3.1 Data-Driven Optimal Control Problem

We consider a continuous-time nonlinear system with the time $t \in \mathbb{R}$ that involves an unknown drift term:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{G}(\boldsymbol{x}(t))\boldsymbol{u}(t). \tag{1}$$

In this system, the control input at t, state at t, unknown drift term, and known input vector field are denoted by $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$, $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$, $\boldsymbol{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, and $\boldsymbol{G} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x\times n_u}$, respectively. The input vector field $\boldsymbol{G}(\boldsymbol{x})$ is regarded as the gain from the input to the dynamics on which this study focuses. The following cost function J is regarded as a measure of the control performance:

$$J(\boldsymbol{u}, \boldsymbol{x}(0)) := \int_0^\infty \left(q(\boldsymbol{x}(t)) + \frac{1}{2}\boldsymbol{u}(t)^\top \boldsymbol{R}(\boldsymbol{x}(t))\boldsymbol{u}(t)\right)\mathrm{d}t. \tag{2}$$

The functions $q : \mathbb{R}^{n_x} \to \mathbb{R}$ and $\boldsymbol{R} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u\times n_u}$ determine the state- and input-dependent costs, respectively. It is desirable for the controller $\boldsymbol{u}$ to minimize the value of the cost $J(\boldsymbol{u}, \boldsymbol{x}(0))$. We assume the following: $\boldsymbol{u}$ is continuous; $\boldsymbol{f}$ and $\boldsymbol{G}$ are locally Lipschitz continuous; $\boldsymbol{f}(0) = 0$ is satisfied; q is positive definite and continuous; $\boldsymbol{R}$ is symmetric positive definite for all $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and continuous; $\boldsymbol{R}^{-1}$ is locally Lipschitz continuous.

While the drift term $\boldsymbol{f}$ is unknown, the training data set is assumed to be provided that contains D pairs of $(\boldsymbol{x}_d, \boldsymbol{f}_d)$ satisfying

$$\boldsymbol{f}_d := \boldsymbol{f}(\boldsymbol{x}_d) + \boldsymbol{\omega}_d, \quad \forall d \in \{1, 2, \ldots, D\}. \tag{3}$$

Assume that, for each d, the noise $\boldsymbol{\omega}_d \in \mathbb{R}^{n_x}$ independently obeys an identical normal distribution with the mean of zero. Such a data set can be acquired by assuming that the values of $\boldsymbol{x}$ and $\mathrm{d}\boldsymbol{x}/\mathrm{d}t - \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$ in (1) are measured.

The purpose of this study is to design a feedback controller for the partially unknown nonlinear system (1) and cost function (2). For the given data set $(\boldsymbol{x}_d, \boldsymbol{f}_d)_{d=1}^D$, we aim to find $\boldsymbol{u}$ minimizing $J(\boldsymbol{u}, \boldsymbol{x}(0))$ over all admissible controllers [24, Sect. 10.1]:

$$\min_{\boldsymbol{u}} J(\boldsymbol{u}, \boldsymbol{x}(0)). \tag{4}$$

It is difficult to minimize $J(\boldsymbol{u}, \boldsymbol{x}(0))$ directly because the drift term $\boldsymbol{f}$ is unknown. We focus on model-based approaches that are efficient in addressing this control problem with an unknown system. The approaches consist of two parts. First, the unknown term $\boldsymbol{f}$ is represented by a mathematical model $\widehat{\boldsymbol{f}}$ using the data set. Second, after replacing $\boldsymbol{f}$ with $\widehat{\boldsymbol{f}}$, solving the optimal control problem provides an optimal controller. The derived controller is approximately optimal to the true term $\boldsymbol{f}$ if $\boldsymbol{f}$ is sufficiently close to $\widehat{\boldsymbol{f}}$. These parts are briefly reviewed in Sects. 3.2 and 3.3.

3.2 Kernel-Based Identification of the Drift Term

We focus on kernel-based system models to identify the unknown system $\boldsymbol{f}$. In the fields of system identification and machine learning [25], the models are promising for describing nonlinear dynamics. GPs [3] and KRR models [7] are well-known examples of such models. The GPs (without variance) and KRR models under technical assumptions are summarized by $\boldsymbol{\mu} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$:

$$\boldsymbol{\mu}(\boldsymbol{x}) := [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_D] \times \left([\boldsymbol{k}_\mathrm{v}(\boldsymbol{x}_1), \ldots, \boldsymbol{k}_\mathrm{v}(\boldsymbol{x}_D)] + \alpha \boldsymbol{I}_D\right)^{-1} \boldsymbol{k}_\mathrm{v}(\boldsymbol{x}), \tag{5}$$

$$\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}) := [k(\boldsymbol{x},\boldsymbol{x}_1),\ldots,k(\boldsymbol{x},\boldsymbol{x}_D)]^\top \in \mathbb{R}^D, \tag{6}$$

where $\alpha > 0 \in \mathbb{R}$ and $k(\boldsymbol{x},\boldsymbol{x}_d) \in \mathbb{R}$ are the hyperparameter and positive definite kernel function, respectively. This study assumes that $k(\boldsymbol{x},\boldsymbol{x}_d)$ is C^2 continuous in $\boldsymbol{x}$, such as a squared exponential kernel.

3.3 Analysis to the Model-Based Optimal Control Problem

We analyze the nonlinear optimal control problem (4) when the unknown term $\boldsymbol{f}(\boldsymbol{x})$ is replaced with the model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$, which indicates the model-based controller design. We assume that $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ is locally Lipschitz and that $\widehat{\boldsymbol{f}}(0) = 0$ is satisfied. A promising tool for analyzing such a problem is the Hamilton-Jacobi-Bellman (HJB) equation [24, Theorem 10.1-2]:

$$\begin{aligned} H(\boldsymbol{x}) &:= \partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \widehat{\boldsymbol{f}}(\boldsymbol{x}) - \frac{1}{2}\partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \boldsymbol{S}(\boldsymbol{x})\partial_{\boldsymbol{x}} V(\boldsymbol{x}) + q(\boldsymbol{x}) \\ &= 0,\ \forall \boldsymbol{x} \in \mathbb{R}^{n_x}, \qquad (7) \\ \boldsymbol{S}(\boldsymbol{x}) &:= \boldsymbol{G}(\boldsymbol{x})\boldsymbol{R}(\boldsymbol{x})^{-1}\boldsymbol{G}(\boldsymbol{x})^\top. \qquad (8) \end{aligned}$$

If there exists a C^1 continuous positive definite solution $V : \mathbb{R}^{n_x} \to \mathbb{R}$ satisfying the HJB equation (7), the optimal controller $\boldsymbol{u}(t) = \boldsymbol{u}_*(\boldsymbol{x}(t))$ to the system model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ is obtained by

$$\boldsymbol{u}_*(\boldsymbol{x}) := -\boldsymbol{R}(\boldsymbol{x})^{-1}\boldsymbol{G}(\boldsymbol{x})^\top \partial_{\boldsymbol{x}} V(\boldsymbol{x}). \tag{9}$$

While the results in [24, Theorem 10.1-2] assumed that $\boldsymbol{R}(\boldsymbol{x})$ is constant, the above-mentioned case using the state-dependent matrix $\boldsymbol{R}(\boldsymbol{x})$ has been easily derived in the same way and has been widely adopted, for example, [26]–[29].

Remark 1 (Difficulty owing to the kernel-based models)**.** *It is difficult to obtain the solution $V(\boldsymbol{x})$ owing to the nonlinearity of $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ and/or $\boldsymbol{G}(\boldsymbol{x})$. If the unknown term $\boldsymbol{f}(\boldsymbol{x})$ is replaced with a kernel-based model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$, such as $\boldsymbol{\mu}(\boldsymbol{x})$ in (5), the controller design suffers from nonlinear kernel functions. The following sections address this difficulty.*

4. Brief Description of the Previous Method [23]

This section reviews our previous method [23] to achieve the goal described by (4). The previous method addressed the difficulty described in Remark 1 under the following assumption:

Assumption 1 (Constant input vector field)**.** *The input vector field in the system* (1) *is a constant matrix $\boldsymbol{G}_{\mathrm{c}} \in \mathbb{R}^{n_x \times n_u}$:*

$$\forall \boldsymbol{x} \in \mathbb{R}^{n_x}, \quad \boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{G}_{\mathrm{c}}. \tag{10}$$

The matrix $\boldsymbol{R}(\boldsymbol{x})$ is assumed to be constant along with Assumption 1. In Sects. 4.1 and 4.2, a novel problem setting and its solution are introduced, respectively.

4.1 Problem Setting to Unify the Identification and Control

Although the kernel-based models $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ have the potential to represent various dynamics, it is not straightforward to solve the HJB equation (7) for such nonlinear models, as mentioned in Remark 1. We introduce a strategy to tackle this drawback, which is unifying the identification and controller design. Let us generalize the kernel-based models as follows:

$$\widehat{\boldsymbol{f}}(\boldsymbol{x}) := \boldsymbol{C}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}), \tag{11}$$

where $\boldsymbol{C} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times D}$ denotes the generalized coefficient matrix. The model in (11) contains some types of functions, such as the GP mean and KRR models $\boldsymbol{\mu}(\boldsymbol{x})$ in (5). If $\boldsymbol{C}(\boldsymbol{x})$ is determined such that we can solve the HJB equation (7), then the optimal controller $\boldsymbol{u}_*(\boldsymbol{x})$ to $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ is easily derived. This strategy suggests solving the following problem.

Main problem: *Find a set $\mathcal{F}$ of kernel-based drift term models $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ defined in* (11) *such that an exact solution $V(\boldsymbol{x})$ to the HJB equation* (7) *is acquired.*

After solving the main problem, we obtain an appropriate $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ included in the set $\mathcal{F}$ to describe $\boldsymbol{f}(\boldsymbol{x})$. Thereafter, the corresponding $V(\boldsymbol{x})$ to the HJB equation (7) is acquired, and the optimal controller $\boldsymbol{u}_*(\boldsymbol{x})$ is provided by (9), which is the objective of our study.

4.2 Limited Solution to the Main Problem

The previous study [23] presented a solution to the main problem under Assumption 1 with constant $\boldsymbol{R}(\boldsymbol{x})$. The way to solve this problem is to decompose the HJB equation (7). Assume that $V(\boldsymbol{x})$ and $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ are parametric functions, and that the HJB equation is decomposed as follows:

$$H(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top \boldsymbol{\Phi}(\boldsymbol{x})^\top \boldsymbol{M}\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}), \tag{12}$$

where $\boldsymbol{M}$ and $\boldsymbol{\Phi}(\boldsymbol{x})$ are a constant matrix and function of $\boldsymbol{x}$, respectively. Free parameters to characterize $V(\boldsymbol{x})$ and $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ are included in $\boldsymbol{M}$. If there exist such a $\boldsymbol{M}$ and $\boldsymbol{\Phi}(\boldsymbol{x})$, satisfying the condition $\boldsymbol{M} = 0$ is reduced to solving the HJB equation as follows:

$$\boldsymbol{M} = 0 \Rightarrow \forall \boldsymbol{x} \in \mathbb{R}^{n_x},\ H(\boldsymbol{x}) = 0. \tag{13}$$

Hence, we derive a set $\mathcal{F}$ of $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ for which we can decompose the HJB equation, as shown in (12).

To realize the above-mentioned strategy, we assume that the solution $V(\boldsymbol{x})$ is a kernel-based function parameterized by

$$V(\boldsymbol{x}) = p[\boldsymbol{\psi}_1(\boldsymbol{x})\boldsymbol{\pi}_1,\ldots,\boldsymbol{\psi}_D(\boldsymbol{x})\boldsymbol{\pi}_D]\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}), \tag{14}$$

where $p \in \mathbb{R}$ and $\boldsymbol{\pi}_d \in \mathbb{R}^{n_\pi}$ are free parameters. The function $\boldsymbol{\psi}_d : \mathbb{R}^{n_x} \to \mathbb{R}^{1\times n_\pi}$ is a predefined C^2 continuous function that satisfies

$$\forall d \in \{1,\dots,D\},\ \boldsymbol{\psi}_d(0) = 0, \tag{15}$$

$$\forall d \in \{1,\dots,D\},\ k(0,\boldsymbol{x}_d)\partial_{\boldsymbol{x}}\boldsymbol{\psi}_d(0) = 0, \tag{16}$$

$$\forall d \in \{1,\dots,D\},\ k(\boldsymbol{x},\boldsymbol{x}_d) = 0 \Rightarrow \boldsymbol{\psi}_d(\boldsymbol{x}) = 0. \tag{17}$$

The positive definiteness of $V(\boldsymbol{x})$ is not guaranteed, whereas it is required for the optimal control design summarized in Sect. 3.3. We aim to ensure this property in a numerical sense when determining the free parameters p and $\boldsymbol{\pi}_d$, which is described in Sect. 5.2. Parametrization using kernel functions has been successfully utilized for some types of functions in control engineering [30]–[34]. Furthermore, functions $V(\boldsymbol{x})$ have been approximated by the sums of basis functions [35], [36].

Let us define $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi D}$, $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^{n_\pi n_x D \times D}$, $\boldsymbol{\Phi}'_d(\boldsymbol{x}) \in \mathbb{R}^{n_x \times n_\pi}$, $\boldsymbol{c}_V(\boldsymbol{x},\boldsymbol{x}_d) \in \mathbb{R}^{n_x}$, and $\boldsymbol{S}_{\mathrm{c}} \in \mathbb{R}^{n_x \times n_x}$ as follows:

$$\boldsymbol{\pi} := [\boldsymbol{\pi}_1{}^\top, \dots, \boldsymbol{\pi}_D{}^\top]^\top, \tag{18}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}) := \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Phi}'_1(\boldsymbol{x})) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{vec}(\boldsymbol{\Phi}'_D(\boldsymbol{x})) \end{bmatrix}, \tag{19}$$

$$\boldsymbol{\Phi}'_d(\boldsymbol{x}) := \boldsymbol{c}_V(\boldsymbol{x},\boldsymbol{x}_d)\boldsymbol{\psi}_d(\boldsymbol{x}) + \partial_{\boldsymbol{x}}\boldsymbol{\psi}_d(\boldsymbol{x}), \tag{20}$$

$$\boldsymbol{c}_V(\boldsymbol{x},\boldsymbol{x}_d) := \begin{cases} \partial_{\boldsymbol{x}} k(\boldsymbol{x},\boldsymbol{x}_d)/k(\boldsymbol{x},\boldsymbol{x}_d) & (k(\boldsymbol{x},\boldsymbol{x}_d) \neq 0) \\ 0 & (k(\boldsymbol{x},\boldsymbol{x}_d) = 0) \end{cases}, \tag{21}$$

$$\boldsymbol{S}_{\mathrm{c}} := \boldsymbol{G}_{\mathrm{c}}\boldsymbol{R}_{\mathrm{c}}^{-1}\boldsymbol{G}_{\mathrm{c}}^\top. \tag{22}$$

where $\boldsymbol{R}_{\mathrm{c}} \succ 0 \in \mathbb{R}^{n_u \times n_u}$ is a positive definite symmetric matrix. The solution to the main problem is derived as follows:

Theorem 1 (Kernel-based HJB equation [23, Theorem 2])**.** *Assume that Assumption 1 and* $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{R}_{\mathrm{c}}$ *hold. For given parameters* $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$ *and* $\boldsymbol{Q} = \boldsymbol{Q}^\top \succ 0 \in \mathbb{R}^{n_x \times n_x}$, *assume that the following relations hold:*

$$\widehat{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}), \tag{23}$$

$$\begin{aligned} q(\boldsymbol{x}) &= \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top\boldsymbol{\Phi}(\boldsymbol{x})^\top(\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x}) \\ &\quad \times \boldsymbol{Q}(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}). \end{aligned} \tag{24}$$

If the equation:

$$\begin{aligned} \boldsymbol{M} &= (\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x})(p\boldsymbol{A} + p\boldsymbol{A}^\top - p^2\boldsymbol{S}_{\mathrm{c}} + 2\boldsymbol{Q})(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x}) \\ &= 0 \in \mathbb{R}^{n_\pi n_x D \times n_\pi n_x D}, \end{aligned} \tag{25}$$

holds, then the HJB equation (7) *holds. Furthermore,* $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ *is locally Lipschitz and* $\widehat{\boldsymbol{f}}(0) = 0$ *is satisfied.*

Remark 2 (Limitation in Theorem 1)**.** *While Theorem 1 obtains the set* $\mathcal{F}$ *using* (23), *it relies on the constant property of the input vector field* $\boldsymbol{G}(\boldsymbol{x})$ *in Assumption 1. However, this indicates that the gain from the control input to the dynamics must be constant, which restricts the range of applications. In the next section, we extend Theorem 1 for state-dependent* $\boldsymbol{G}(\boldsymbol{x})$.

5. Proposed Method for State-Dependent Input Vector Fields

In this section, our previous method [23] is extended to overcome the main problem without the limitation of Assumption 1. We can employ the state-dependent input vector field $\boldsymbol{G}(\boldsymbol{x})$, as well as the constant vector field for enhancing the applicability of the previous method. Section 5.1 provides the main results in this study. The implementation of the proposed method is provided in Sect. 5.2. Section 5.3 extends the results for robust nonlinear control.

5.1 Solution to the Main Problem

We involve the state-dependent vector field $\boldsymbol{G}(\boldsymbol{x})$ by decomposing $\boldsymbol{S}(\boldsymbol{x})$ into nonlinear scalar functions and constant matrices. Because of the local Lipschitz continuity of $\boldsymbol{G}(\boldsymbol{x})$ and $\boldsymbol{R}(\boldsymbol{x})^{-1}$, any $\boldsymbol{S}(\boldsymbol{x})$ can be represented by

$$\boldsymbol{S}(\boldsymbol{x}) = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{R}(\boldsymbol{x})^{-1}\boldsymbol{G}(\boldsymbol{x})^\top = \sum_{i=1}^{N} \xi_i(\boldsymbol{x})\boldsymbol{S}_i, \tag{26}$$

where $\xi_i : \mathbb{R}^{n_x} \to \mathbb{R}$, $\boldsymbol{S}_i \in \mathbb{R}^{n_x \times n_x}$, and N are a locally Lipschitz scalar function of $\boldsymbol{x}$, constant matrix, and positive integer, respectively. Using this decomposition, we derive the following theorem to solve the main problem even for the state-dependent vector field $\boldsymbol{G}(\boldsymbol{x})$ in (26).

Theorem 2 **(Extended kernel-based HJB equation).** *Assume that* (26) *holds. For given parameters* $\boldsymbol{A}_i \in \mathbb{R}^{n_x \times n_x}$ *and* $\boldsymbol{Q}_i \in \mathbb{R}^{n_x \times n_x}$ *(*$i = 1,2,\dots,N$*), assume that* $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ *and* $q(\boldsymbol{x})$ *are given by*

$$\widehat{\boldsymbol{f}}(\boldsymbol{x}) = \left(\sum_{i=1}^{N} \xi_i(\boldsymbol{x})\boldsymbol{A}_i\right)(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}), \tag{27}$$

$$\begin{aligned} q(\boldsymbol{x}) &= \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top\boldsymbol{\Phi}(\boldsymbol{x})^\top(\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x}) \\ &\quad \times \left(\sum_{i=1}^{N} \xi_i(\boldsymbol{x})\boldsymbol{Q}_i\right)(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}). \end{aligned} \tag{28}$$

If the equation:

$$\begin{aligned} \boldsymbol{M}_i &= (\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x})(p\boldsymbol{A}_i + p\boldsymbol{A}_i{}^\top - p^2\boldsymbol{S}_i + 2\boldsymbol{Q}_i) \\ &\quad \times (\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x}) \\ &= 0 \in \mathbb{R}^{n_\pi n_x D \times n_\pi n_x D}, \end{aligned} \tag{29}$$

holds for all i, *then the HJB equation* (7) *holds. Furthermore,* $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ *is locally Lipschitz and* $\widehat{\boldsymbol{f}}(0) = 0$ *is satisfied.*

Proof. The proof is described in Appendix A. □

Remark 3 (Contribution of Theorem 2)**.** *Theorem 2 provides a solution to the main problem, even if the input vector field* $\boldsymbol{G}(\boldsymbol{x})$ *depends on the state* $\boldsymbol{x}$. *The set* $\mathcal{F}$ *is determined by* (27) *for which an exact solution to the HJB equation* (7) *is obtained. Furthermore, Theorem 2 covers the existing* *results in Theorem 1 by setting $N = 1$, $\xi_1(\boldsymbol{x}) = 1$, and $\boldsymbol{S}_1 = \boldsymbol{S}_{\mathrm{c}}$.*

Remark 4 (How to satisfy the algebraic equation (29))**.** *The algebraic matrix equation* (29) *must hold to satisfy the HJB equation* (7)*. A condition to satisfy the matrix equation is provided by the setting:*

$$\boldsymbol{Q}_i = (p^2\boldsymbol{S}_i - p\boldsymbol{A}_i - p\boldsymbol{A}_i{}^{\top})/2. \tag{30}$$

Remark 5 (Positive definite properties)**.** *Recall that the positive definiteness of $q(\boldsymbol{x})$ and $V(\boldsymbol{x})$ is not guaranteed in Theorem 2. We aim to ensure this property in a numerical sense when determining the free parameters, as described in the next subsection.*

Remark 6 (Optimality to the true system)**.** *We investigate the optimality of the model-based controller $\boldsymbol{u}_*(\boldsymbol{x}(t))$ in* (9) *to true drift term $\boldsymbol{f}(\boldsymbol{x})$ by analyzing the HJB equation* (7)*. Let $H^{\boldsymbol{f}}(\boldsymbol{x})$ be the function $H(\boldsymbol{x})$ of the HJB equation to $\boldsymbol{f}(\boldsymbol{x})$. The controller is optimal to the true system if a C^1 continuous positive definite function $V(\boldsymbol{x})$ satisfies $H^{\boldsymbol{f}}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$. In this sense, we regard $H^{\boldsymbol{f}}(\boldsymbol{x})$ as a metric of the optimality. Our analysis in Appendix B indicates that $H^{\boldsymbol{f}}(\boldsymbol{x})$ is close to zero, i.e., the optimality is enhanced, by decreasing the modeling error $\|\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x})\|$.*

5.2 Implementation

In this subsection, we implement the main results of Theorem 2 by modifying the implementation in our previous work [23, Sect. V], where differences between the two implementations are described in the last of this subsection. We obtain an appropriate $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ contained in the set $\mathcal{F}$ that obeys (27) such that the algebraic equation (29) holds under the settings (14), (28), and (30). This process reduces to an optimization of the free parameters, p, $\boldsymbol{A}_i$, and $\boldsymbol{\pi}$. Solving the optimization obtains the model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ and controller $\boldsymbol{u}_*(\boldsymbol{x})$ in (9) simultaneously because the HJB equation (7) is solved.

The details of the optimization are described herein, where the functions depending on the parameters are expressed by $\widehat{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$, $V(\boldsymbol{x};p,\boldsymbol{\pi})$, $q(\boldsymbol{x};p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$, and $\boldsymbol{u}_*(\boldsymbol{x};p,\boldsymbol{\pi})$ with the definition of $\boldsymbol{A}_{\mathrm{all}} := [\boldsymbol{A}_1,\ldots,\boldsymbol{A}_N]$. We consider the following constrained minimization:

$$\begin{aligned}&\min_{p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi}} g(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})\\ &\text{s.t. } \forall d,\ \ V(\widetilde{\boldsymbol{x}}_d;p,\boldsymbol{\pi}) \geq 0,\ \ q(\widetilde{\boldsymbol{x}}_d;p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})) \geq 0,\end{aligned} \tag{31}$$

where $\widetilde{\boldsymbol{x}}_d$ $(d = 1,2,\ldots,\widetilde{D})$ is a predefined point. The objective function $g(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$ is defined as follows:

$$\begin{aligned}g(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi}) := \Big(\sum_{d=1}^{\widetilde{D}} \|\widehat{\boldsymbol{f}}(\widetilde{\boldsymbol{x}}_d;\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi}) - \boldsymbol{\mu}(\widetilde{\boldsymbol{x}}_d)\|^2\Big)\\ + w_{\boldsymbol{\pi}}\|\boldsymbol{\pi}\|^2 + w_p p^2 + w_A\|\mathrm{vec}(\boldsymbol{A}_{\mathrm{all}})\|^2.\end{aligned} \tag{32}$$

where $w_{\boldsymbol{\pi}} > 0$, $w_p > 0$, and $w_A > 0$ are coefficients for regularizing the norms of the parameters. The function $\boldsymbol{\mu}(\boldsymbol{x})$ is the kernel-based model defined in (5) in Sect. 3.2. Minimizing $g(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$ enhances the identification accuracy because the difference between $\widehat{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$ and $\boldsymbol{\mu}(\boldsymbol{x})$ is reduced. The constraints in (31) are employed to satisfy the positive definite properties of $q(\boldsymbol{x};p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$ and $V(\boldsymbol{x};p,\boldsymbol{\pi})$ in a numerical sense. Note that $\boldsymbol{S}_i$ is defined such that (26) hold before solving the minimization problem. The parameter $\boldsymbol{Q}_i$ in (30) is determined using the obtained p, $\boldsymbol{A}_{\mathrm{all}}$, and $\boldsymbol{S}_i$.

The optimization in (31) involves a large number of parameters because the dimension of $\boldsymbol{\pi}$ is proportional to the number of data points D. We employ the technique in our previous work [23, Sect. V] to reduce the number of parameters. This technique approximates an optimal $\boldsymbol{\pi}$ to (31) using the following function $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})$:

$$\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}}) \in \arg\min_{\boldsymbol{\pi}} g(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi}), \tag{33}$$

This approximation is applicable because $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})$ is provided in explicit form as follows.

Proposition 1 (Explicit minimizer). *Assume that* (27) *holds. The function $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})$ in* (33) *is uniquely expressed as:*

$$\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}}) = (\boldsymbol{Y}(\boldsymbol{A}_{\mathrm{all}})^{\top}\boldsymbol{Y}(\boldsymbol{A}_{\mathrm{all}}) + w_{\boldsymbol{\pi}}\boldsymbol{I}_{n_\pi D})^{-1}\boldsymbol{Y}(\boldsymbol{A}_{\mathrm{all}})^{\top}\boldsymbol{y}, \tag{34}$$

where $\boldsymbol{Y}(\boldsymbol{A}_{\mathrm{all}}) \in \mathbb{R}^{n_x\widetilde{D}\times n_\pi D}$ and $\boldsymbol{y} \in \mathbb{R}^{n_x\widetilde{D}}$ are defined as

$$\boldsymbol{Y}(\boldsymbol{A}_{\mathrm{all}}) := \sum_{i=1}^{N}(\boldsymbol{I}_{\widetilde{D}} \otimes \boldsymbol{A}_i)\boldsymbol{Y}_i \tag{35}$$

$$\boldsymbol{Y}_i := \begin{bmatrix}\boldsymbol{Y}_{i,1,1} & \cdots & \boldsymbol{Y}_{i,1,D}\\ \vdots & & \vdots\\ \boldsymbol{Y}_{i,\widetilde{D},1} & \cdots & \boldsymbol{Y}_{i,\widetilde{D},D}\end{bmatrix}, \tag{36}$$

$$\boldsymbol{Y}_{i,s,d} := \xi_i(\widetilde{\boldsymbol{x}}_s)k(\widetilde{\boldsymbol{x}}_s,\boldsymbol{x}_d)\boldsymbol{\Phi}'_d(\widetilde{\boldsymbol{x}}_s) \in \mathbb{R}^{n_x\times n_\pi}, \tag{37}$$

$$\boldsymbol{y} := \begin{bmatrix}\boldsymbol{\mu}(\widetilde{\boldsymbol{x}}_1)\\ \vdots\\ \boldsymbol{\mu}(\widetilde{\boldsymbol{x}}_{\widetilde{D}})\end{bmatrix}. \tag{38}$$

Proof. The proof is described in Appendix C. □

Along with the replacement of $\boldsymbol{\pi}$ with $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})$, the constrained minimization (31) is relaxed as the following unconstrained minimization:

$$\min_{p,\boldsymbol{A}_{\mathrm{all}}} g(p,\boldsymbol{A}_{\mathrm{all}},\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})) + b(p,\boldsymbol{A}_{\mathrm{all}},\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\mathrm{all}})), \tag{39}$$

The function $b(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})$ is defined as follows:

$$\begin{aligned}b(p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi}) := \sum_{d=1}^{\widetilde{D}}\Big(w_V\ln(1+\exp(-w_{\mathrm{e}}V(\widetilde{\boldsymbol{x}}_d;p,\boldsymbol{\pi})))\\ + w_q\ln(1+\exp(-w_{\mathrm{e}}q(\widetilde{\boldsymbol{x}}_d;p,\boldsymbol{A}_{\mathrm{all}},\boldsymbol{\pi})))\Big),\end{aligned} \tag{40}$$

Algorithm 1 Design of optimal controllers for kernel-based models

```
Input: the training data set $(\boldsymbol{x}_d, \boldsymbol{f}_d)_{d=1}^{D}$, weight $\boldsymbol{R}$, predefined points $\widetilde{\boldsymbol{x}}_d$
    ($d = 1, 2, \dots, \widetilde{D}$), functions $\xi_i(\boldsymbol{x})$ and matrices $\boldsymbol{S}_i$ ($i = 1, 2, \dots, N$)
Output: the drift term model $\widehat{\boldsymbol{f}}(\boldsymbol{x}; \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$ and model-based optimal controller $\boldsymbol{u}_*(\boldsymbol{x}; p, \boldsymbol{\pi})$
1: Obtain the GP mean model $\boldsymbol{\mu}(\boldsymbol{x})$ in (5) by utilizing the data set
2: Define the function $\boldsymbol{\psi}_d(\boldsymbol{x})$ according to the conditions (15)–(17)
3: Obtain the values of $p$ and $\boldsymbol{A}_{\rm all}$ by solving (39)
4: Obtain the values of $\boldsymbol{\pi} = \overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all})$ in (34)
5: Obtain $\widehat{\boldsymbol{f}}(\boldsymbol{x}; \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$, $V(\boldsymbol{x}; p, \boldsymbol{\pi})$, $q(\boldsymbol{x}; p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$ in (27), (14), and (28) with (30)
6: Obtain $\boldsymbol{u}_*(\boldsymbol{x}; p, \boldsymbol{\pi})$ in (9) with the substitution of $V(\boldsymbol{x}; p, \boldsymbol{\pi})$
```

where $w_V \geq 0$ and $w_q \geq 0$, and $w_{\rm e} > 0$ are the coefficients. The function $b(p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$ consists of the softplus penalty functions that indicate $\ln(1 + \exp(a)) \approx \max\{0, a\}$ for $a \in \mathbb{R}$ [37]. Such functions help in satisfying the positive definiteness because the penalties occur for approximately negative $V(\widetilde{\boldsymbol{x}}_d; p, \boldsymbol{\pi}))$ and $q(\widetilde{\boldsymbol{x}}_d; p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi}))$.

Finally, we summarize the proposed method in Algorithm 1. The GP mean model $\boldsymbol{\mu}(\boldsymbol{x})$ is obtained in Line 1. In Lines 3 and 4, all the parameters p, $\boldsymbol{A}_{\rm all}$, and $\boldsymbol{\pi}$ are determined. Thereafter, the model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ and controller $\boldsymbol{u}_*(\boldsymbol{x})$ in (9) are obtained because the HJB equation (7) is solved.

Algorithm 1 is equivalent to the algorithm of our previous method [23]. The extension from the previous method to the proposed method is to modify the functions $g(p, \boldsymbol{A}_{\rm all}, \overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all})) + b(p, \boldsymbol{A}_{\rm all}, \overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all}))$ and $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all})$ in Proposition 1.

Remark 7 (The degree of freedom in choosing $q(\boldsymbol{x})$). *The function $q(\boldsymbol{x})$ depends on the free parameters p, $\boldsymbol{A}_{\rm all}$, and $\boldsymbol{\pi}$. If a certain target function $q_*(\boldsymbol{x})$ is given, it is possible to determine these parameters such that $q(\boldsymbol{x}; p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$ is close to $q_*(\boldsymbol{x})$ by solving the minimization problem* (31) *with adding the term $w_{q2} \sum_{d=1}^{\widetilde{D}} \|q(\widetilde{\boldsymbol{x}}_d; p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi}) - q_*(\widetilde{\boldsymbol{x}}_d)\|^2$ in the objective function $g(p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$, where $w_{q2} > 0$ is a weight coefficient. This term indicates a distance between $q(\boldsymbol{x}; p, \boldsymbol{A}_{\rm all}, \boldsymbol{\pi})$ and $q_*(\boldsymbol{x})$.*

5.3 Extension for Nonlinear Robust Control

The main results presented in Sect. 5.1 are extended to a robust control problem called H_∞ control [38] in a manner similar to that in our previous work [23]. This subsection considers the case in which the system (1) involves a disturbance $\boldsymbol{w}(t) \in \mathbb{R}^{n_w}$ and the performance output $z(t)$:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{G}(\boldsymbol{x}(t))\boldsymbol{u}(t) + \boldsymbol{G}_{\rm d}(\boldsymbol{x}(t))\boldsymbol{w}(t), \tag{41}$$

$$z(t) := \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}(t)) \\ \boldsymbol{R}_{\rm sqr}(\boldsymbol{x}(t))\boldsymbol{u}(t) \end{bmatrix} \in \mathbb{R}^{n_h + n_u}, \tag{42}$$

We assume the following: $\boldsymbol{G}_{\rm d} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_w}$ is locally Lipschitz; $\sup_t \|\boldsymbol{w}(t)\| < \infty$ is satisfied; $\boldsymbol{w}(t)$ is locally Lipschitz, nonzero, and square-integrable, implying $0 < \int_0^\infty \|\boldsymbol{w}(t)\|^2 {\rm d}t < \infty$; $\boldsymbol{h} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_h}$ and $\boldsymbol{R}_{\rm sqr} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u \times n_u}$ are continuous; $q(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^{\rm T}\boldsymbol{h}(\boldsymbol{x})/2$ and $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{R}_{\rm sqr}(\boldsymbol{x})^{\rm T}\boldsymbol{R}_{\rm sqr}(\boldsymbol{x})$ hold.

The H_∞ control problem focuses on a L_2 gain from $\boldsymbol{w}$ to z. We attempt to find a feedback controller satisfying the condition that the L_2 gain for any $\boldsymbol{w}$ is smaller than or equal to a given parameter $\gamma \in (0, \infty)$ for $\boldsymbol{x}(0) = 0$:

$$\left(\int_0^\infty \|z(t)\|^2 {\rm d}t\right)^{\frac{1}{2}} \Big/ \left(\int_0^\infty \|\boldsymbol{w}(t)\|^2 {\rm d}t\right)^{\frac{1}{2}} \leq \gamma, \tag{43}$$

It is well known that the controller design corresponds to solving the Hamilton-Jacobi-Isaacs (HJI) equation [24, Theorem 10.3-1], [39], [40]. In the model-based approach, that is, $\boldsymbol{f}$ is replaced with $\widehat{\boldsymbol{f}}$, the HJI equation is reduced to the HJB equation (7) that uses the following definition instead of the definition (8):

$$\boldsymbol{S}(\boldsymbol{x}) := \boldsymbol{G}(\boldsymbol{x})\boldsymbol{R}(\boldsymbol{x})^{-1}\boldsymbol{G}(\boldsymbol{x})^{\rm T} - \frac{1}{\gamma^2}\boldsymbol{G}_{\rm d}(\boldsymbol{x})\boldsymbol{G}_{\rm d}(\boldsymbol{x})^{\rm T}. \tag{44}$$

If a C^1 continuous positive definite solution $V : \mathbb{R}^{n_x} \to \mathbb{R}$ to the HJI equation exists, $\boldsymbol{u}_*(\boldsymbol{x})$ in (9) corresponds to the controller realizing (43) when $\boldsymbol{f}$ is replaced with $\widehat{\boldsymbol{f}}$. Hence, we can exploit Theorem 2 to the H_∞ control problem by using the definition (44).

Theorem 3 (Extended kernel-based HJI equation). *Assume that* (44) *holds instead of* (8) *and that all the assumptions in Theorem 2 are satisfied. The statements in Theorem 2 hold if the HJB equation* (7) *is replaced with the HJI equation.*

Proof. This is proved in the same way to Theorem 2. □

6. Numerical Examples

This section demonstrates the method proposed in Sect. 5. Section 6.1 formulates a practical example of the unknown system (1) and describes the simulation setting. Section 6.2 shows the simulation results.

6.1 Target System and Simulation Setting

Consider the target nonlinear system (1) with nonlinear external forces and a nonlinear input vector field, as illustrated in Fig. 2. Let the system state denotes the position $x_{\rm m}$ in [m] and velocity $\dot{x}_{\rm m}$ in [m/s] of the mass point at the time t in [s]:

$$\boldsymbol{x} := \begin{bmatrix} x_{\rm m} \\ \dot{x}_{\rm m} \end{bmatrix}, \tag{45}$$

where $x_{\rm m}$ is assumed to be included in $(-X_{\rm m}, X_{\rm m})$ with a distance $X_{\rm m}$. For the flexible arm, the mass is negligible, and the length varies depending on $x_{\rm m}$ without any resistance. The point with the mass $M_{\rm m}$ is assumed to move only in the $x_{\rm m}$-direction without any friction. The control input is considered as the torque $\boldsymbol{u}$ in [N·m] that affects the acceleration of the mass point. The $x_{\rm m}$-directional force of the torque is

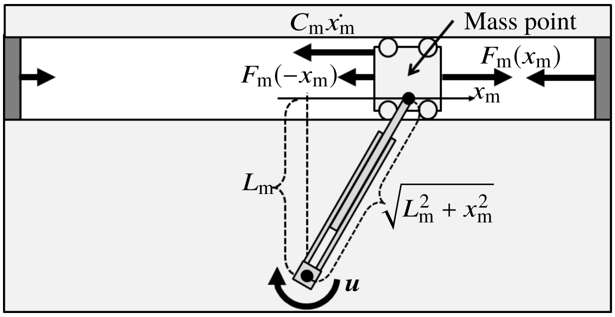

$C_{\mathrm{m}}\dot{x}_{\mathrm{m}}$
Mass point
$F_{\mathrm{m}}(x_{\mathrm{m}})$
$F_{\mathrm{m}}(-x_{\mathrm{m}})$
x_{m}
L_{m}
$\sqrt{L_{\mathrm{m}}^2 + x_{\mathrm{m}}^2}$
$\boldsymbol{u}$

Fig. 2 Practical example of the system with a state-dependent input vector field.

formulated by:

$$\frac{L_{\mathrm{m}}}{\sqrt{L_{\mathrm{m}}^2 + x_{\mathrm{m}}^2}} \frac{\boldsymbol{u}}{\sqrt{L_{\mathrm{m}}^2 + x_{\mathrm{m}}^2}} = \frac{L_{\mathrm{m}}}{(L_{\mathrm{m}}^2 + [\boldsymbol{x}]_1^2)}\boldsymbol{u}, \tag{46}$$

where L_{m} is the length of the arm at $x_{\mathrm{m}} = 0$. The external forces $F_{\mathrm{m}}(x_{\mathrm{m}})$, such as magnetic forces, are assumed to depend on the position x_{m}:

$$F_{\mathrm{m}}(x_{\mathrm{m}}) := \frac{A_{\mathrm{m}}}{(X_{\mathrm{m}} - x_{\mathrm{m}})^2}. \tag{47}$$

The x_{m}-directional viscose resistance is given by $C_{\mathrm{m}}\dot{x}_{\mathrm{m}}$ with the coefficient C_{m}. Based on these physical models, the target system (1) is constructed as follows:

$$\boldsymbol{f}(\boldsymbol{x}) := \begin{bmatrix} [\boldsymbol{x}]_2 \\ (F_{\mathrm{m}}([\boldsymbol{x}]_1) - F_{\mathrm{m}}(-[\boldsymbol{x}]_1) - C_{\mathrm{m}}[\boldsymbol{x}]_2)/M_{\mathrm{m}} \end{bmatrix}, \tag{48}$$

$$\boldsymbol{G}(\boldsymbol{x}) := \begin{bmatrix} 0 \\ L_{\mathrm{m}}/(M_{\mathrm{m}}(L_{\mathrm{m}}^2 + [\boldsymbol{x}]_1^2)) \end{bmatrix}. \tag{49}$$

The physical parameters are set to $M_{\mathrm{m}} = 0.2$ [kg], $X_{\mathrm{m}} = 4$ [m], $A_{\mathrm{m}} = 1.6$ [N/m^2], $L_{\mathrm{m}} = 5$ [m], and $C_{\mathrm{m}} = 0.1$ [N·s/m].

Recall that $\boldsymbol{f}(\boldsymbol{x})$ is unknown, whereas its training data set is provided in our problem setting. In the setting of the data set $(\boldsymbol{x}_d, \boldsymbol{f}_d)_{d=1}^D$ with $D = 81$, the states $\boldsymbol{x}_d$ are equivalent to all members of the set $\{-2, -1.5, \ldots, 2\} \times \{-2, -1.5, \ldots, 2\}$. The drift terms $\boldsymbol{f}_d$ obeys (3), wherein the noise $\boldsymbol{\omega}_d$ for each d follows the normal distribution independently with the mean of zero and covariance of $0.2^2\boldsymbol{I}_2$.

Algorithm 1 is implemented as follows. The weight in the cost function is set to $\boldsymbol{R} = 1$. The function $\boldsymbol{S}(\boldsymbol{x})$ in (26) is decomposed using the following setting with $N = 1$:

$$\xi_1(\boldsymbol{x}) = [\boldsymbol{G}(\boldsymbol{x})]_2^2, \tag{50}$$

$$\boldsymbol{S}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{51}$$

In Line 1, the GPML package [41] is employed for the GP regression. The GP mean model $\boldsymbol{\mu}(\boldsymbol{x})$ is developed utilizing the data set and the square exponential kernel function:

$$k(\boldsymbol{x}, \boldsymbol{x}_d) := \beta \exp\left(\frac{-1}{2}(\boldsymbol{x} - \boldsymbol{x}_d)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{x}_d)\right), \tag{52}$$

Table 1 Mean values and standard deviations of the root mean square errors for the 100 training data sets.

	Mean value	Standard deviation
First component $[\widehat{\boldsymbol{f}}(\boldsymbol{x})]_1$	0.078	0.019
Second component $[\widehat{\boldsymbol{f}}(\boldsymbol{x})]_2$	0.094	0.023

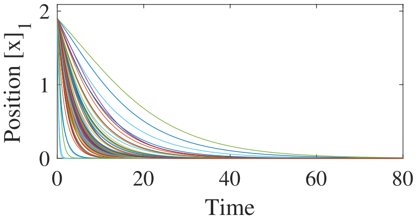
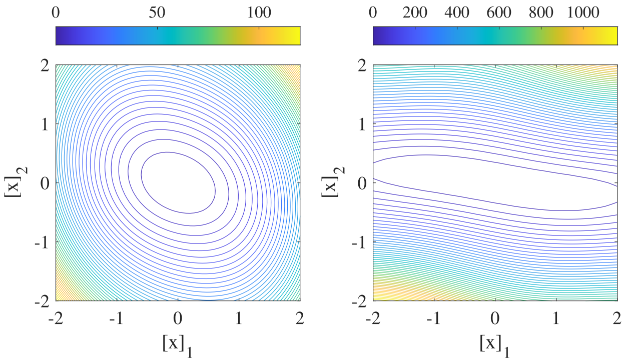


Fig. 3 Control results for the different training data sets.

where $\beta > 0 \in \mathbb{R}$ and diagonal $\boldsymbol{\Sigma} > 0 \in \mathbb{R}^{n_x \times n_x}$ are the hyperparameters. Further details of the regression are found in [23, Sections V.A and VI.A]. In Line 2, the function is set as $\boldsymbol{\psi}_d(\boldsymbol{x}) := [([\boldsymbol{x}]_1)^2, ([\boldsymbol{x}]_2)^2, [\boldsymbol{x}]_1[\boldsymbol{x}]_2]$. In Lines 3 and 4, the coefficients in (32) and (40) are set as $w_V = 1000$, $w_q = 1000$, $w_{\mathrm{e}} = 10$, $w_{\boldsymbol{\pi}} = 5$, $w_p = 0.001$, and $w_A = 0.001$. The points $\widetilde{\boldsymbol{x}}_d$ are sampled on $[-2, 2] \times [-2, 2]$ at regular intervals, where $\widetilde{D} = 121$. The minimization problem in (39) is numerically solved using the quasi-Newton (BFGS) method [42, Chapter 6]. To avoid numerical errors due to a significantly large value of $\exp(-w_{\mathrm{e}}V(\widetilde{\boldsymbol{x}}_d))$ in (40), we use the transformation of $\ln(1+\exp(-w_{\mathrm{e}}V(\widetilde{\boldsymbol{x}}_d)) = V' + \ln(\exp(-V') + \exp(-w_{\mathrm{e}}V(\widetilde{\boldsymbol{x}}_d) - V'))$ with $V' := \max\{-w_{\mathrm{e}}V(\widetilde{\boldsymbol{x}}_d), 0\}$. The term $\exp(-w_{\mathrm{e}}q(\widetilde{\boldsymbol{x}}_d))$ is transformed in the same way. The initial values of the decision variables are set to $p = 3$, $\boldsymbol{A}_1 = (p^2\boldsymbol{S}_1 - 2\boldsymbol{Q}_1)/2p$, and $\boldsymbol{Q}_1 = \boldsymbol{I}_2 > 0$. The simulation is implemented utilizing the Dormand-Prince method [43], where the sampling time and time horizon are set to 0.005 and 80, respectively.

6.2 Numerical Results

We generated 100 different training data sets $(\boldsymbol{x}_d, \boldsymbol{f}_d)_{d=1}^D$ by changing the random noise $(\boldsymbol{\omega}_d)_{d=1}^D$ in the data sets. We evaluated the proposed method for the 100 data sets. The mean values and standard deviations of the root mean square errors of the proposed model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ are listed in Table 1. The mean values and deviations were smaller than 5% of the magnitude (approximately 2.0) of $\boldsymbol{f}(\boldsymbol{x})$. Figure 3 shows the state trajectories by applying the proposed controller, where the multiple lines indicate the results for the different data sets. The states for all the results approached around the origin, indicating that the system was successfully controlled. Figure 4 represents the positive definite properties of $V(\boldsymbol{x}; p, \boldsymbol{\pi})$ and $q(\boldsymbol{x}; p, \boldsymbol{A}_{\mathrm{all}}, \boldsymbol{\pi})$ for one of the data sets. The positive definiteness of both the function was numerically verified, which was assumed in the proposed method. Based on these results, we confirmed that the proposed method performs well for a class of unknown systems.



(a) Plot of $V(\boldsymbol{x}; p, \boldsymbol{\pi})$ (b) Plot of $q(\boldsymbol{x}; p, \boldsymbol{A}_{\text{all}}, \boldsymbol{\pi})$

Fig. 4 Contour lines of $V(\boldsymbol{x}; p, \boldsymbol{\pi})$ and $q(\boldsymbol{x}; p, \boldsymbol{A}_{\text{all}}, \boldsymbol{\pi})$ obtained for a training data set.

7. Conclusion

In this study, we solved a data-driven optimal control problem for partially unknown nonlinear systems by extending our previous work [23]. The target system contains a state-dependent input vector field whereas the previous work did not focus on such a system. The proposed method obtains a kernel-based model and model-based optimal controller simultaneously. The unknown drift term is described by the model using a training data set. The nonlinear optimal control problem for the model is solved through an analytical approach; thus, simplifying the controller design. The effectiveness of the method was confirmed through a numerical simulation.

The proposed approach should be extended to include the variance of GP models in future work. Extensions for other control problems and practical applications will also be considered.

Acknowledgments

This study was partly supported by JSPS KAKENHI Grant Number JP18K04222.

References

[1] T. Nishi, P. Doshi, and D. Prokhorov, "Merging in congested freeway traffic using multipolicy decision making and passive actor-critic learning," IEEE Trans. Intell. Veh., vol.4, no.2, pp.287–297, 2019.

[2] M. Torchio, N.A. Wolff, D.M. Raimondo, L. Magni, U. Krewer, R.B. Gopaluni, J.A. Paulson, and R.D. Braatz, "Real-time model predictive control for the optimal charging of a lithium-ion battery," Proc. 2015 Annual American Control Conf., pp.4536–4541, 2015.

[3] C.E. Rasmussen and C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, Massachusetts, 2006.

[4] B. Hu, G. Su, J. Jiang, J. Sheng, and J. Li, "Uncertain prediction for slope displacement time-series using Gaussian process machine learning," IEEE Access, vol.7, pp.27535–27546, 2019.

[5] W. Guo, T. Pan, Z. Li, and S. Chen, "Model calibration method for soft sensors using adaptive Gaussian process regression," IEEE Access, vol.7, pp.168436–168443, 2019.

[6] Y. Ito, K. Fujimoto, Y. Tadokoro, and T. Yoshimura, "On stabilizing control of Gaussian processes for unknown nonlinear systems," Preprints of the 20th IFAC World Congress, pp.15955–15960, 2017.

[7] V. Vovk, "Kernel ridge regression," Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, pp.105–116, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2013.

[8] M.P. Deisenroth, C.E. Rasmussen, and J. Peters, "Gaussian process dynamic programming," Neurocomputing, vol.72, no.7-9, pp.1508–1524, 2009.

[9] P. Hemakumara and S. Sukkarieh, "Learning UAV stability and control derivatives using Gaussian processes," IEEE Trans. Robot., vol.29, no.4, pp.813–824, 2013.

[10] M.P. Deisenroth, D. Fox, and C.E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.2, pp.408–423, 2015.

[11] F. Xie, W. Hong, W. Wu, K. Liang, and C. Qiu, "Current distribution method of induction motor for electric vehicle in whole speed range based on Gaussian process," IEEE Access, vol.7, pp.165974–165984, 2019.

[12] K. Chen, J. Yi, and D. Song, "Gaussian processes model-based control of underactuated balance robots," Proc. 2019 International Conf. on Robotics and Automation, pp.4458–4464, 2019.

[13] J. Kocijan, R. Murray-Smith, C.E. Rasmussen, and B. Likar, "Predictive control with Gaussian process models," Proc. IEEE Region 8 EUROCON 2003. Computer as a Tool., 2003.

[14] A. Grancharova, J. Kocijan, and T.A. Johansen, "Explicit stochastic nonlinear predictive control based on Gaussian process models," Proc. 2007 European Control Conf., 2007.

[15] G. Cao, Gaussian Process based Model Predictive Control, Ph.D. thesis, Massey University, 2017.

[16] L. Hewing, J. Kabzan, and M.N. Zeilinger, "Cautious model predictive control using Gaussian process regression," IEEE Trans. Control Syst. Technol., vol.28, no.6, pp.2736–2743, 2020.

[17] T.X. Nghiem, "Linearized Gaussian processes for fast data-driven model predictive control," Proc. 2019 American Control Conference, pp.1629–1634, 2019.

[18] E. Bradford, L. Imsland, and E.A. del Rio-Chanona, "Nonlinear model predictive control with explicit back-offs for Gaussian process state space models," Proc. IEEE 58th Conf. on Decision and Control, pp.4747–4754, 2019.

[19] Y. Pan and E.A. Theodorou, "Data-driven differential dynamic programming using Gaussian processes," Proc. 2015 Annual American Control Conf., pp.4467–4472, 2015.

[20] J. Boedecker, J.T. Springenberg, J. Wulfing, and M. Riedmiller, "Approximate real-time optimal control based on sparse Gaussian process models," Proc. 2014 IEEE Symp. on Adaptive Dynamic Programming and Reinforcement Learning, 2014.

[21] Y. Ito, K. Fujimoto, and Y. Tadokoro, "Second-order bounds of Gaussian kernel-based functions and its application to nonlinear optimal control with stability," arXiv:1707.06240v1, 2017.

[22] F. Berkenkamp, M. Turchetta, A.P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," Proc. Advances in Neural Information Processing Systems 30, pp.908–918, 2017.

[23] Y. Ito, K. Fujimoto, and Y. Tadokoro, "Kernel-based Hamilton–Jacobi equations for data-driven optimal and H-infinity control," IEEE Access, vol.8, pp.131047–131062, 2020.

[24] F.L. Lewis, D.L. Vrabie, and V.L. Syrmos, Optimal Control, 3rd ed., John Wiley & Sons, Hoboken, New Jersey, 2012.

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

[26] J.J. Murray, C.J. Cox, G.G. Lendaris, and R. Saeks, "Adaptive dynamic programming," IEEE Trans. Systems, Man, and Cybern. C, vol.32, no.2, pp.140–153, 2002.

[27] Y. Huang and W.M. Lu, "Nonlinear optimal control: Alternatives to Hamilton-Jacobi equation," Proc. IEEE 35th Conf. on Decision and Control, pp.3942–3947, 1996.

[28] D. McCaffrey and S.P. Banks, "Geometric existence theory for the

control-affine nonlinear optimal regulator," J. Math. Anal. Appl., vol.305, no.1, pp.380–390, 2005.

[29] C.H. Won and S. Biswas, "Optimal control using an algebraic method for control-affine non-linear systems," Int. J. Control, vol.80, no.9, pp.1491–1502, 2007.

[30] E. Todorov, "Eigenfunction approximation methods for linearly-solvable optimal control problems," Proc. 2009 IEEE Symp. on Adaptive Dynamic Programming and Reinforcement Learning, 2009.

[31] R. Kamalapurkar, J.A. Rosenfeld, and W.E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," Automatica, vol.74, pp.247–258, 2016.

[32] P. Giesl, "Construction of a local and global Lyapunov function for discrete dynamical systems using radial basis functions," J. Approx. Theory, vol.153, no.2, pp.184–211, 2008.

[33] P. Giesl and S. Hafstein, "Review on computational methods for Lyapunov functions," Discrete and Continuous Dynamical Systems - Series B, vol.20, no.8, pp.2291–2331, 2015.

[34] J. Bouvrie and B. Hamzi, "Kernel methods for the approximation of some key quantities of nonlinear systems," J. Computational Dynamics, vol.4, no.1, pp.1–19, 2017.

[35] R.W. Beard, G. Saridis, and J. Wen, "Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation," J. Optimiz. Theory Appl., vol.96, pp.589–626, 1998.

[36] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming," IEEE Trans. Cybern., vol.48, no.2, pp.500–509, 2018.

[37] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," Advances in Neural Information Processing Systems, pp.451–457, 2000.

[38] A.J. van der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control," IEEE Trans. Autom. Control, vol.37, no.6, pp.770–784, 1992.

[39] D. McCaffrey, "Geometric existence theory for the control-affine H_∞ problem," J. Math. Anal. Appl., vol.324, no.1, pp.682–695, 2006.

[40] J. Morimoto and K. Doya, "Robust reinforcement learning," Advances in Neural Information Processing Systems, pp.1061–1067, 2001.

[41] C.E. Rasmussen and H. Nickisch, Gaussian Process Regression and Classification Toolbox version 4.2, 2018.

[42] J. Nocedal and S.J. Wright, Numerical Optimization, 2nd ed., Springer Science+Business Media, LLC, New York, 2006.

[43] J.R. Dormand and P.J. Prince, "A family of embedded Runge-Kutta formulae," J. Comput. Appl. Math., vol.6, pp.19–26, 1980.

Appendix A: Proof of Theorem 2

The partial derivative $\partial_{\boldsymbol{x}} V(\boldsymbol{x})$ is given by [23, Lemma 1]:

$$\partial_{\boldsymbol{x}} V(\boldsymbol{x}) = p(\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}). \tag{A$\cdot$1}$$

Using this expression, we obtain the following relations:

$$\begin{aligned}
&\partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \widehat{\boldsymbol{f}}(\boldsymbol{x})\\
&= \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top \boldsymbol{\Phi}(\boldsymbol{x})^\top (\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x}) p\\
&\quad \times \Big(\sum_{i=1}^{N} \xi_i(\boldsymbol{x}) \boldsymbol{A}_i \Big) (\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x}) \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}),
\end{aligned} \tag{A$\cdot$2}$$

$$\begin{aligned}
&\partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \boldsymbol{S}(\boldsymbol{x}) \partial_{\boldsymbol{x}} V(\boldsymbol{x})\\
&= \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top \boldsymbol{\Phi}(\boldsymbol{x})^\top (\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x}) p^2\\
&\quad \times \Big(\sum_{i=1}^{N} \xi_i(\boldsymbol{x}) \boldsymbol{S}_i \Big) (\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x}) \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}).
\end{aligned} \tag{A$\cdot$3}$$

Substituting these relations into the HJB equation (7), we obtain the first statement:

$$\begin{aligned}
H(\boldsymbol{x}) &:= \frac{1}{2} \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top \boldsymbol{\Phi}(\boldsymbol{x})^\top (\boldsymbol{\pi} \otimes \boldsymbol{I}_{n_x})\\
&\quad \times \Big(\sum_{i=1}^{N} \xi_i(\boldsymbol{x}) \Big(p\boldsymbol{A}_i + p{\boldsymbol{A}_i}^\top - p^2 \boldsymbol{S}_i + 2\boldsymbol{Q}_i \Big) \Big)\\
&\quad \times (\boldsymbol{\pi}^\top \otimes \boldsymbol{I}_{n_x}) \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})\\
&= \frac{1}{2} \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})^\top \boldsymbol{\Phi}(\boldsymbol{x})^\top \Big(\sum_{i=1}^{N} \xi_i(\boldsymbol{x}) \boldsymbol{M}_i \Big) \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x}).
\end{aligned} \tag{A$\cdot$4}$$

Next, the second statement is proved in a manner similar to that in [23, Appendix B]. Because $\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\mathrm{v}}(\boldsymbol{x})$ and $\xi_i(\boldsymbol{x})$ are locally Lipschitz, $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ expressed in (27) is locally Lipschitz. The property $\widehat{\boldsymbol{f}}(0) = 0$ holds because of the conditions (15)–(17). The proof has been completed.

Appendix B: Optimality Analysis of the Proposed Controller

After the HJB equation $H(\boldsymbol{x}) = 0$ to the drift term model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ is satisfied by the proposed method, $H^{\boldsymbol{f}}(\boldsymbol{x})$ is given using the modeling error $(\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x}))$:

$$\begin{aligned}
&H^{\boldsymbol{f}}(\boldsymbol{x})\\
&:= \partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \boldsymbol{f}(\boldsymbol{x}) - \frac{1}{2} \partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top \boldsymbol{S}(\boldsymbol{x}) \partial_{\boldsymbol{x}} V(\boldsymbol{x}) + q(\boldsymbol{x})\\
&= H(\boldsymbol{x}) + \partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top (\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x}))\\
&= \partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top (\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x})),
\end{aligned} \tag{A$\cdot$5}$$

Let us define $\|\boldsymbol{f}(\cdot)\|_{s,\mathcal{X}} := (\int_{\mathcal{X}} \|\boldsymbol{f}(\boldsymbol{x})\|^s \mathrm{d}\boldsymbol{x})^{1/s}$, which is the norm of $\boldsymbol{f}(\cdot)$ over a subset $\mathcal{X} \subset \mathbb{R}^{n_x}$ for $s \in \{1, 2\}$. We assume that there exist scalars $C_{\partial V} > 0$ and $\delta > 0$ such that $\|\partial_{\boldsymbol{x}} V(\cdot)\|_{2,\mathcal{X}} < C_{\partial V}$ holds if $\|\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot)\|_{2,\mathcal{X}} < \delta$ holds. Then, we obtain the following relation using Cauchy-Schwarz inequality:

$$\begin{aligned}
&\|\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot)\|_{2,\mathcal{X}} < \delta\\
&\Rightarrow \|H^{\boldsymbol{f}}(\cdot)\|_{1,\mathcal{X}}\\
&\quad = \|\partial_{\boldsymbol{x}} V(\cdot)^\top (\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot))\|_{1,\mathcal{X}}\\
&\quad = \int_{\mathcal{X}} |\partial_{\boldsymbol{x}} V(\boldsymbol{x})^\top (\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x}))| \mathrm{d}\boldsymbol{x}\\
&\quad \le \int_{\mathcal{X}} \|\partial_{\boldsymbol{x}} V(\boldsymbol{x})\| \|\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x})\| \mathrm{d}\boldsymbol{x}\\
&\quad \le \Big(\int_{\mathcal{X}} \|\partial_{\boldsymbol{x}} V(\boldsymbol{x})\|^2 \mathrm{d}\boldsymbol{x} \Big)^{1/2} \Big(\int_{\mathcal{X}} \|\boldsymbol{f}(\boldsymbol{x}) - \widehat{\boldsymbol{f}}(\boldsymbol{x})\|^2 \mathrm{d}\boldsymbol{x} \Big)^{1/2}\\
&\quad \le C_{\partial V} \|\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot)\|_{2,\mathcal{X}},
\end{aligned} \tag{A$\cdot$6}$$

This implies that the norm of the optimality metric $H^{\boldsymbol{f}}(\cdot)$ is linearly bounded by the modeling error, i.e., $\|H^{\boldsymbol{f}}(\cdot)\|_{1,\mathcal{X}} = O(\|\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot)\|_{2,\mathcal{X}})$ as $\|\boldsymbol{f}(\cdot) - \widehat{\boldsymbol{f}}(\cdot)\|_{2,\mathcal{X}} \to 0$. This result justifies developing the data-driven model $\widehat{\boldsymbol{f}}(\boldsymbol{x})$ with high accuracy.

Appendix C: Proof of Proposition 1

We prove the statement by extending the existing results [23, Proposition 1, Appendix E]. Using the existing result transforms $\widehat{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{A}_{\rm all},\boldsymbol{\pi})$ as follows:

$$\begin{aligned}\widehat{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{A}_{\rm all},\boldsymbol{\pi}) &= \Big(\sum_{i=1}^{N}\xi_i(\boldsymbol{x})\boldsymbol{A}_i\Big)(\boldsymbol{\pi}^\top\otimes\boldsymbol{I}_{n_x})\boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{k}_{\rm v}(\boldsymbol{x})\\ &= \sum_{i=1}^{N}\xi_i(\boldsymbol{x})\boldsymbol{A}_i\sum_{d=1}^{D}\boldsymbol{\Phi}'_d(\boldsymbol{x})\boldsymbol{\pi}_d k(\boldsymbol{x},\boldsymbol{x}_d)\\ &= \sum_{i=1}^{N}\boldsymbol{A}_i[\xi_i(\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{x}_1)\boldsymbol{\Phi}'_1(\boldsymbol{x}),\ldots\\ &\qquad \ldots,\xi_i(\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{x}_D)\boldsymbol{\Phi}'_D(\boldsymbol{x})]\boldsymbol{\pi}. \qquad (\mathrm{A}\cdot 7)\end{aligned}$$

Using this expression, we obtain

$$\begin{aligned}\begin{bmatrix}\widehat{\boldsymbol{f}}(\widetilde{\boldsymbol{x}}_1;\boldsymbol{A}_{\rm all},\boldsymbol{\pi})\\ \vdots\\ \widehat{\boldsymbol{f}}(\widetilde{\boldsymbol{x}}_{\widetilde{D}};\boldsymbol{A}_{\rm all},\boldsymbol{\pi})\end{bmatrix} &= \sum_{i=1}^{N}\begin{bmatrix}\boldsymbol{A}_i[\boldsymbol{Y}_{i,1,1},\ldots,\boldsymbol{Y}_{i,1,D}]\boldsymbol{\pi}\\ \vdots\\ \boldsymbol{A}_i[\boldsymbol{Y}_{i,\widetilde{D},1},\ldots,\boldsymbol{Y}_{i,\widetilde{D},D}]\boldsymbol{\pi}\end{bmatrix}\\ &= \sum_{i=1}^{N}\begin{bmatrix}\boldsymbol{A}_i & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \boldsymbol{A}_i\end{bmatrix}\boldsymbol{Y}_i\boldsymbol{\pi}\\ &= \boldsymbol{Y}(\boldsymbol{A}_{\rm all})\boldsymbol{\pi}. \qquad (\mathrm{A}\cdot 8)\end{aligned}$$

Substituting the relation (A· 8) into $g(p,\boldsymbol{A}_{\rm all},\boldsymbol{\pi})$ in (33) provides the well-known regularized quadratic minimization with a constant $C_{\boldsymbol{\pi}}$:

$$\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all}) \in \arg\min_{\boldsymbol{\pi}}\Big(\|\boldsymbol{Y}(\boldsymbol{A}_{\rm all})\boldsymbol{\pi}-\boldsymbol{y}\|^2 + w_{\boldsymbol{\pi}}\|\boldsymbol{\pi}\|^2 + C_{\boldsymbol{\pi}}\Big). \qquad (\mathrm{A}\cdot 9)$$

Thus, the solution $\overline{\boldsymbol{\pi}}(\boldsymbol{A}_{\rm all})$ for this minimization is uniquely obtained by (34), where using the condition $w_{\boldsymbol{\pi}} > 0$ ensures the nonsingularity of the matrix $(\boldsymbol{Y}(\boldsymbol{A}_{\rm all}))^\top\boldsymbol{Y}(\boldsymbol{A}_{\rm all})+w_{\boldsymbol{\pi}}\boldsymbol{I}_{n_\pi D})$. This completes the proof.

Yuji Ito received his B.S., M.S., and Ph.D. degrees from Nagoya University, Aichi, Japan, in 2009, 2011, and 2014, respectively. Since 2011, he has been with TOYOTA CENTRAL R&D LABS., INC., Japan. His research interests include stochastic systems theory and nonlinear optimal control. Dr. Ito is a member of Institute of Electrical and Electronics Engineers and the Society of Instrument and Control Engineers.

Kenji Fujimoto received the B.S. and M.S. degrees in engineering and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 1994, 1996, and 2001, respectively. He is currently a Professor with the Graduate School of Engineering, Kyoto University, Kyoto, Japan. From 1997 to 2004 he was a Research Associate of Graduate School of Engineering and Graduate School of Informatics, Kyoto University, Japan. From 2004 to 2012 he was an Associate Professor of Graduate School of Engineering, Nagoya University, Japan. From 1999 to 2000 he was a Research Fellow of Department of Electrical Engineering, Delft University of Technology, The Netherlands. He has held visiting research positions at the Australian National University, Australia and Delft University of Technology, The Netherlands in 1999 and 2002, respectively. He received the IFAC Congress Young Author Prize in IFAC World Congress 2005 and SICE Control Division Pioneer Award in 2007. His research interests include nonlinear control and stochastic systems theory.