Algebraic design of physical computing system

Komatsu, Mizuka
Yaguchi, Takaharu
Nakajima, Kohei

(Citation)
Physica D: Nonlinear Phenomena,470(A):134382

(Issue Date)
2024-12

(Resource Type)
journal article

(Version)
Version of Record

Physica D 470 (2024) 134382

Contents lists available at ScienceDirect

Physica D

journal homepage: www.elsevier.com/locate/physd

Algebraic design of physical computing system

Mizuka Komatsu [a,*], Takaharu Yaguchi [b], Kohei Nakajima [c]

[a] Graduate School of System Informatics, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo, 657-8501, Japan
[b] Graduate School of Science, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo, 657-8501, Japan
[c] Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

ARTICLE INFO

Communicated by Stefan Wiggins

Keywords:
Physical computing system
Input–output equation
Applied algebra
Gröbner basis

ABSTRACT

Recently, computational techniques that employ physical systems (physical computing systems) have been developed. To utilize physical computing systems, their design strategy is important. Although there are practical learning-based methods and theoretical approaches, no general method exists that provides specific design guidelines for given systems with rigorous theoretical support. In this paper, we propose a novel algebraic design framework for a physical computing system, which is capable of extracting specific design guidelines. Our approach describes input–output relationships algebraically and relates them to given target tasks. Two theorems are presented in this paper. The first theorem offers a basic strategy for algebraic design. The second theorem explores the "replaceability" of such systems. Their possible implementations are investigated through experiments. In particular, the design of inputs of a system so that it generates multiple target time-series and the replacement of stationary or non-stationary target systems by a given system that is designed algebraically are included. The proposed framework is shown to have the potential of designing given physical computing systems with theoretical support.

1. Introduction

Recent developments in sensing and Internet of Things technology require enormous complex time-series data to be processed efficiently in real-time. The von Neumann architecture, which is a conventional computational architecture that aligns the processor and memory separately, causes an intrinsic limitation in processing speed, the von Neumann bottleneck. To design a successful real-time information processing method with lower computational and energy costs, it is necessary to reconsider the computational architecture and concept [1]. Therefore, many approaches based on the non-von Neumann type architectures have recently been proposed, capitalizing on the power of material properties [2,3] and the diverse dynamics of physical systems [4]. Physical systems outsource the computational load to natural physical dynamics and add some properties in addition to computation, such as robustness against a radioactive environment [5,6], which significantly extends the application domains. In this paper, the term systems refers to physical computing systems, and we propose a design principle for the systems based on the theory of algebra.

We focus on mapping input sequences to output sequences, which we consider computing in this paper. In the field of physical computing and its related domains, the terms "time-series" and "sequence" are used to refer to both continuous and discrete cases [7–11]. In the following, "time-series" refers specifically to continuous time-series, while discrete time-series are referred to as "discrete time-series." In this paper, we examine physical computing, where a physical system characterized by its parameters is employed to map an input sequence (Fig. 1). In the research field of physical computing, designing a system that exhibits desirable computing has been a focus of attention [12–15]. Common and practical methods include simulation- [12] and learning-based methods [13]. Although the efficacy of such methods has been partially confirmed, they basically rely on an enormous number of trial and error attempts or simulations, and/or it is necessary to learn their input–output relationships, which are not supported by theories. On one hand, researchers [14,16] have attempted to understand the design principle of physical computing systems, which are based on the theory of nonlinear filter approximation [17]. In these studies, computational tasks are reduced to the approximation of a class of filters by the systems. In other words, each filter in the class has the role of mapping an input stream to an output stream. Then, sufficient conditions for the approximation such as the echo-state property [18,19], fading memory property and point separation property [14,17] are provided. However, the practical use of such theories can be challenging. Specifically, it is difficult to design physical computing systems so that they satisfy these conditions; specific design guidelines for the systems based on the nonlinear filter approximation theory are difficult to obtain. There

* Corresponding author.
E-mail addresses: m-komatsu@bear.kobe-u.ac.jp (M. Komatsu), yaguchi@pearl.kobe-u.ac.jp (T. Yaguchi), k-nakajima@isi.imi.i.u-tokyo.ac.jp (K. Nakajima).

https://doi.org/10.1016/j.physd.2024.134382
Received 27 March 2024; Received in revised form 13 September 2024; Accepted 15 September 2024
Available online 1 October 2024

is another direction for theoretically investigating physical computing systems, which is based on static feedback linearizability [15,20]. Static feedback linearizability is a well established tool in the field of nonlinear control theory, targeting systems described in state-space representation. Roughly speaking, a static feedback linearizable system is a system such that it can be transformed into a linear system by a change of coordinates and a static feedback control. In [15,20], mass–spring systems are targeted and their feedback linearizablity are demonstrated. Based on this, such systems of dimension N, with an appropriate change of coordinates and feedback control, are shown to be able to enumerate solutions of Nth order differential equations in response to an external input. Since the proof of feedback linearizability of the systems in [15,20] is constructive, one can say that these theories provide specific design guidelines, the feedback controls in these case, in contrast to the theories based on nonlinear filter approximation [14,16]. However, a class of static feedback linearizable system is limited and hence the theoretical frameworks provided in [15,20] are only applicable to a limited class of physical computing systems. In the field of nonlinear control theory, there is a property of system called differential flatness [21–23], which is closely related to feedback liniearizability. Roughly speaking, differentially flat systems can be transformed into linear systems with a change of coordinates and a dynamic feedback control [24]. Thus, the differential flatness is a generalized concepts of feedback linearization which has a potential to extend the theories such as [15,20]. However, the existence of a computable criterion so as to decide if a system is differentially flat remains open [24]. Thus, challenges for existing approaches in designing physical computing systems can be organized as follows:

- The theoretical approaches for physical computing system based on nonlinear filter approximation [14,16] do not provide specific design methods.
- Practical simulation- or learning-based approaches [12,13] may provide specific design guidelines for physical computing systems, however, they do not have theoretical guarantees.
- The theoretical methods based on static feedback linearizability [15,20] allow us to obtain specific design guidelines of physical computing systems. However, their applicability is limited to a subclass of physical computing systems.
- The applicability of the approaches [15,20] may be extended by introducing differential flatness [21–23]. However, determining whether a system is differentially flat in a systematic manner remains an open problem.

Based on the above, we aim at establishing a theoretical framework for designing physical computing systems that can offer specific design guidelines based on algebraic theory and related computational techniques. We assume that a physical computing system is modeled as a state-space model and examine its specific design guidelines via state variables elimination supported by algebraic theory. An overview of our framework is depicted in Fig. 1. In this framework, the specific input–output relationships of a given system, $F = 0$ in Fig. 1, are first described via elimination of the state variables. Then, F is related to the computational tasks (Theorems 1 and 2). To be more precise, the input–output relationships are characterized by the Gröbner basis, which is an algebraic technique. This makes it possible to describe the relationships and obtain design guidelines using computer algebra software without being aware of the underlying algebraic theory. In contrast to existing theories based on nonlinear filter approximation [14,16] or feedback linearization [15,20], our approach does not essentially limit the class of physical computing systems while extracting specific design guidelines. This is because, our approach relies on $F = 0$, which is explicitly computable for any physical computing systems described by polynomial state-space models. In addition, it is worth to mention that our approach has the relation to differential flatness as demonstrated in Section 4.2. This suggests that our approach could serve as a clue for further developing existing theories.

In this study, two main theorems are provided. Theorem 1 outlines a fundamental design strategy for computational tasks. Such a process for adjustments is often conducted empirically in practical methods [12,13]. We demonstrate how the proposed strategy works when designing inputs in Experiments 1 and 2. Theorem 2 extends Theorem 1 by addressing situations where one physical computing system might be preferable over another, considering the "replaceability" of systems. As an example, such a situation may arise as a result of the emergence of diverse implementations of physical computing. Experiment 3 demonstrates the effectiveness of Theorem 2, which provides specific guidelines for replacing one system with another. The limitations of our framework are discussed in Section 5.

2. Algebraic preliminaries

2.1. Algebraic terminology

As preparation, we introduce some algebraic terminology related to the study. For basic terminology concerning the field, ring, polynomial ring, and order, please refer to [25,26]. Throughout this study, $K[x]$ denotes a polynomial ring whose variables are $x = (x_1, \dots, x_s)$ over a field K.

Definition 1 (*Ideal on a Polynomial Ring*)**.** A non-empty subset I of $K[x]$ satisfies

$$\text{if } p \in I, q \in I, \text{ then } p + q \in I, \tag{1}$$

$$\text{if } p \in I, q \in K[x], \text{ then } qp \in I, \tag{2}$$

where $K[x]$ is a polynomial ring; p, q are polynomials.

Definition 2 (*Generating Set*)**.** For an ideal I, suppose that there exists a non-zero subset $\{p_\lambda \mid \lambda \in \Lambda\}$ of ring $K[x]$ that satisfies

$$I = \left\{ \sum_{\lambda \in \Lambda} q_\lambda p_\lambda \,\middle|\, \text{for all } q_\lambda \in K[x] \right\}. \tag{3}$$

Then, $\{p_\lambda \mid \lambda \in \Lambda\}$ is defined as a generating set of I. $p_\lambda (\lambda \in \Lambda)$ is defined as a generator of I.

Notably, the set on the right side of (3) is shown to satisfy the definition of the ideal [25]. Representing the ideal I, whose generator is $\{p_\lambda \mid \lambda \in \Lambda\}$, is commonly expressed as follows:

$$I = \langle \{p_\lambda \mid \lambda \in \Lambda\} \rangle. \tag{4}$$

The underlying algorithm for deriving the key equation $F = 0$ in Fig. 1, called the input–output equation, is polynomial division. When performing division between the polynomials of a single variable, in general, we first arrange the terms of the polynomial in descending order of the degree of the variable. We then focus on the monomials whose variable degree is the highest in the polynomial to be divided and for the division. If the monomial for the polynomial to be divided is higher than or equal to that of the polynomial to divide, then the polynomial to be divided is factorized by the highest degree term of the polynomial to divide. Therefore, the order of monomials is critical in dividing polynomials of a single variable. This holds for the multivariate case as well [25]. In general, the monomial ordering fixes the order of the monomials in the set of monomials of interest [25]. Below, we consider the monomial order to handle polynomials, which is a special type of the total order.

Definition 3 (*Total Order*)**.** $\leq$ is a total order on a set Σ if the followings holds for all $a, b, c \in \Sigma$:

$$a \leq a, \tag{5}$$

$$a \leq b \text{ or } b \leq a, \tag{6}$$

$$\text{if } a \leq b \text{ and } b \leq a, \text{ then } a = b, \tag{7}$$

$$\text{if } a \leq b \text{ and } b \leq c, \text{ then } a \leq c. \tag{8}$$

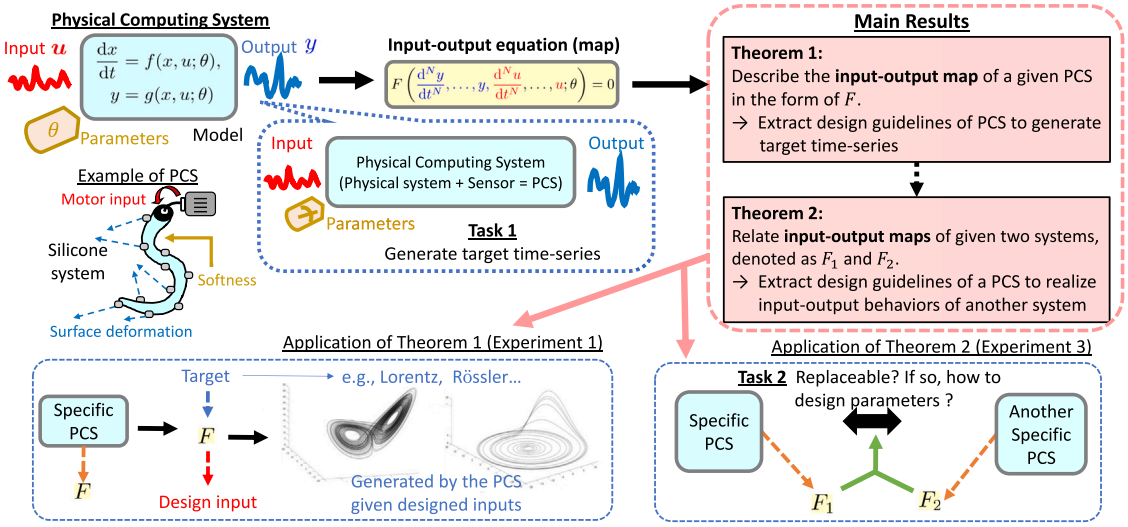

Fig. 1. Overview of the proposed framework. The framework of a physical computing system is depicted at the upper left. Theorems on the algebraic design of physical computing systems are summarized in the upper right, which corresponds to the main results. Experiments 1 and 3 are illustrated at the bottom.

Conventionally, we denote $a < b$ if both $a \leq b$ and $a \neq b$ hold. Among a set Σ equipped with a total order, we can compare every pair of elements. By adding some properties to the total order, the monomial orderings are defined as follows.

Definition 4 (*Monomial Order*)**.** Monomial order $<$ is the total order on a set of monomials M_s of a polynomial ring such that

$$1 < u, \text{for all } u \in M_s, u \neq 1, \tag{9}$$

$$\text{if } u \leq v\, u, v \in M_s \text{ then } uw \leq vw, \text{for all } w \in M_s. \tag{10}$$

Definition 5 (*Lexcographic Order*)**.** Let us consider a ring $K[x]$ and monomial order $<$ on the ring such that for arbitrary $u = x_1^{\alpha_1} x_2^{\alpha_2}, \dots, x_s^{\alpha_s}$, $v = x_1^{\beta_1} x_2^{\beta_2}, \dots$, and $x_s^{\beta_s} \in K[x]$, $u < v$ holds if the left-most element among the non-zero elements of $(\beta_1 - \alpha_1, \beta_2 - \alpha_2, \dots, \beta_s - \alpha_s)$ is positive. We define $<$ as the lexcographic order.

The lexicographic order can be used to eliminate variables from ideals [25]; hence, it is used in this study. To eliminate variables, To eliminate variables, we perform computation of the Gröbner basis, whose underlying algorithm is polynomial division. Before explaining this, we introduce related terminology.

Definition 6. The largest monomial in the set of monomials, whose coefficients are non-zero, appearing in a polynomial $p \in K[x]$ with respect to a given monomial order $<$ is defined as the leading monomial of p. We denote this by $\mathrm{LM}_<(p)$.

Definition 7. The coefficient of the leading monomial of $p \in K[x]$ with respect to a given monomial order $<$ is defined as the leading coefficient of p. We denote this as $\mathrm{LC}_<(p)$.

Definition 8. Let us consider a ring $K[x]$ with a given a monomial order $<$. Suppose we have non-zero polynomials p and h in $K[x]$. We represent h as

$$h = qp + R, \tag{11}$$

where the followings holds:

- if $R \neq 0$, then R is not represented by $R = q'p$, where $q' \in K[x]$;
- if $q \neq 0$, then $\mathrm{LM}_<(h) \geq \mathrm{LM}_<(qp)$.

R is defined as the remainder of h divided by p with respect to $<$ over $K[x]$.

The existence of q and R in Definition 8 is shown [25, p. 64]. Although this is the case of a polynomial being divided by a single polynomial, it can be extended to polynomial division using a set of polynomials [25]. See [25] for the details and algorithms on the division of polynomials of multiple variables. In general, because the division algorithm is implemented on computer algebra software, R can be specifically computed using a given h, p, and monomial order. Because $\mathrm{LM}_<(h) \geq \mathrm{LM}_<(qp)$ holds if $q \neq 0$, $\mathrm{LM}_<(h) < \mathrm{LM}_<(qp)$ implies $q = 0$, which leads to an identical equation $h = R$. That is, if the leading monomial of a polynomial to be used to divide is greater than that of the polynomial to be divided, the remainder is trivial.

3. Problem setting

In this study, we assume that the physical computing system is modeled by a system of differential equations with an observation function as follows:

$$\frac{dx}{dt} = f(x,u;\theta), \tag{12a}$$

$$y = g(x,u;\theta) \tag{12b}$$

where $f(\cdot,\cdot;\theta) : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$, $g(\cdot,\cdot;\theta) : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}$. $x \in \mathbb{R}^N$ is the state variable vector of the physical computing system; $u \in \mathbb{R}^M$ is the input vector; $y \in \mathbb{R}^M$ is the output variable; $\theta \in \mathbb{R}^n$ is the parameter vector. Let us denote the initial state variable vector as $x(0) \in \mathbb{R}^N$. The system of differential Eqs. (12a) describes the physical system of interest. We assume the existence and uniqueness of its solution.

An observation equation (12b) describes how to observe the output of the system. In the field of reservoir computing [27], this is commonly referred to as the readout, where the outputs (12b) correspond to the outputs from readouts equipped with a physical reservoir.

To employ the algebraic methods, throughout this paper, we assume that f and g are polynomials of x, u, whose coefficients are rational functions of θ; $f(x,u;\theta), g(x,u;\theta) \in \mathbb{C}(\theta)[x,u]$. Hereafter, symbols with a hat denote outputs from the physical computing system and those with a bar or tilde denote the given time-series.

To design physical computing system, we first consider whether the output of a given system generates target time-series $\bar{y}(t)$.

Target task 1 (*Time-Series Generation*)**.** *Let us denote output $y(t)$ of (12b) as $\hat{y}(t;\theta,u,x(0))$ given $\theta \in \Theta, x(0) \in \mathbb{R}^N$, and u in a set of functions, $\mathbb{R} \to \mathbb{R}^M$, denoted as U. Suppose that we have a target time-series denoted as $\bar{y}(t)$ over a time duration. A time-series generation task is to find $(\theta,u,x(0)) \in \mathbb{R}^n \times U \times \mathbb{R}^N$ such that $\hat{y}(t;\theta,u,x(0)) = \bar{y}(t)$ holds over the specified domain.*

As mentioned in the previous section, various implementations of physical computing have been proposed using diverse physical systems and materials. Therefore, a situation in which computing with a certain physical computing system is preferable over another may arise. The following task that considers such a situation can be successfully handled using our algebraic approaches, as shown in Section 4.3.

Target task 2 (*Designing a System and Exploiting Its Replaceability with Another System*)**.** *Suppose that we have two systems described by state-space models such as (12a), (12b). The first is a system to be designed, whereas the other is the target. With the same input $u \in U$, their output variables are denoted as $\hat{y}(t;\theta,u,x(0))$ and $\bar{y}(t;\tilde{\theta},u,\tilde{x}(0))$ given initial states $x(0),\tilde{x}(0)\in\mathbb{R}^N$ and parameter vectors $\theta\in\mathbb{R}^n,\tilde{\theta}\in\mathbb{R}^{\tilde{n}}$, respectively. In this task, $(\theta,x(0))\in\mathbb{R}^n\times\mathbb{R}^N$ is to be found, if it exists, so that $\hat{y}(t;\theta,u,x(0))=\bar{y}(t;\tilde{\theta},u,\tilde{x}(0))$ holds over the specified duration. Note that if such $(\theta,x(0))$ exists, the system is said to be replaceable with respect to the target system.*

We make two assumptions regarding target time-series $\bar{y}(t)$. The first relates to its smoothness (Assumption 1), which is assumed throughout this paper. See Section 5 for a discussion related to this assumption. The other assumption (Assumption 2) is introduced in Section 4.3 and is related to the replaceability.

Assumption 1. Target time-series $\bar{y}(t)$ is a sufficiently smooth function from $\mathbb{R}$ to $\mathbb{R}$ over the duration under consideration.

Assumption 2. Let $\mathcal{F}_u$ denote a set of sufficiently smooth functions containing functions from $\mathbb{R}$ to $\mathbb{R}^M$. Given and fixed $u\in\mathcal{F}_u$ and $\tilde{\theta}\in\mathbb{R}^{\tilde{n}}$, target time-series $\bar{y}(t)$ satisfies $h\left(\frac{\mathrm{d}^N\bar{y}}{\mathrm{d}t^N},\dots,\bar{y},\frac{\mathrm{d}^N u}{\mathrm{d}t^N},\dots,u;\tilde{\theta}\right)=0$ where h is a polynomial differential equations up to the Nth order of $\bar{y}$.

4. Proposed framework

4.1. Key ingredient in algebraic design: the input–output map of physical computing systems

The most important idea in the proposed framework is to extract the input–output relations of systems by using an algebraic method. To extract equations representing the input–output relations in the system modeled by (12a) and (12b), we consider eliminating its state variables. Proposition 1 shown in [28] below states that this elimination is certainly possible. Also, because (12a) contains derivatives of the state variables, such eliminations are realized through algebraic manipulations and derivations of the model described by (12a), (12b). Here, the question is how many times we need to differentiate the models to extract equations representing the input–output relations, which may not be bounded.

Before showing the precise statement of Proposition 1, we introduce some algebraic notation. In terms of algebra, we regard the model as polynomials of the states, inputs, outputs, and their higher-order derivatives. To deal with (12a) and (12b) as polynomial equations, we introduce a notation of variables with a superscript non-negative integer i in parentheses $x^{(i)}$, and replace $\frac{\mathrm{d}^i x}{\mathrm{d}t^i}$ with it. The same notation is introduced for y and u. The variable correspondence is summarized in Table 1. Note that $x^{(i)}$ and $x^{(j)}$ are distinct variables if $i\neq j$. Introducing the notation in the right-sides of (12a), (12b), we obtain $f(x^{(0)},u^{(0)};\theta),g(x^{(0)},u^{(0)};\theta)$. Polynomial equations corresponding to (12a) and (12b) are represented as follows:

$$x^{(1)}=f^{(0)}(x^{(0)},u^{(0)};\theta),\quad f^{(0)}(x^{(0)},u^{(0)};\theta)\in\mathbb{C}(\theta)[x^{(0)},u^{(0)}], \tag{13a}$$

$$y^{(0)}=g^{(0)}(x^{(0)},u^{(0)};\theta),\quad g^{(0)}(x^{(0)},u^{(0)};\theta)\in\mathbb{C}(\theta)[x^{(0)},u^{(0)}], \tag{13b}$$

where the symbols f,g are replaced with $f^{(0)},g^{(0)}$. Although the elements of $(x,\frac{\mathrm{d}x}{\mathrm{d}t},y,u)$ depend on t by definition of (12a), (12b), the ones in $(x^{(1)},x^{(0)},y^{(0)},u^{(0)})$ does not, and thus, (13a), (13b) are regarded as polynomial equations; $x^{(1)}-f^{(0)},y-g^{(0)}\in\mathbb{C}(\theta)[x^{(1)},x^{(0)},y^{(0)},u^{(0)}]$.

Table 1
Correspondence of variables in differential equations (12a), (12b) and polynomial equations (13a), (13b). i denotes a non-negative integer.

Variables in (12a), (12b)	Variables in (13a), (13b)
$\frac{\mathrm{d}^i x}{\mathrm{d}t^i}$	$x^{(i)}$
$\frac{\mathrm{d}^i y}{\mathrm{d}t^i}$	$y^{(i)}$
$\frac{\mathrm{d}^i u}{\mathrm{d}t^i}$	$u^{(i)}$

In the following, we consider differential equations and polynomial equations that are obtained by differentiating (12a), (12b) with respect to t up to Nth order:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}x\\ \frac{\mathrm{d}x}{\mathrm{d}t}\\ \vdots\\ \frac{\mathrm{d}^N x}{\mathrm{d}t^N}\end{pmatrix}=\begin{pmatrix}\frac{\mathrm{d}x}{\mathrm{d}t}\\ \frac{\mathrm{d}^2 x}{\mathrm{d}t^2}\\ \vdots\\ \frac{\mathrm{d}^{N+1}x}{\mathrm{d}t^{N+1}}\end{pmatrix}=\begin{pmatrix}f(x,u;\theta)\\ \frac{\mathrm{d}}{\mathrm{d}t}f(x,u;\theta)\\ \vdots\\ \frac{\mathrm{d}^N}{\mathrm{d}t^N}f(x,u;\theta)\end{pmatrix}, \tag{14a}$$

$$\begin{pmatrix}y\\ \frac{\mathrm{d}y}{\mathrm{d}t}\\ \vdots\\ \frac{\mathrm{d}^N y}{\mathrm{d}t^N}\end{pmatrix}=\begin{pmatrix}g(x,u;\theta)\\ \frac{\mathrm{d}}{\mathrm{d}t}g(x,u;\theta)\\ \vdots\\ \frac{\mathrm{d}^N}{\mathrm{d}t^N}g(x,u;\theta)\end{pmatrix}. \tag{14b}$$

Here, $\frac{\mathrm{d}^i}{\mathrm{d}t^i}f(x,u;\theta)$, $\frac{\mathrm{d}^i}{\mathrm{d}t^i}g(x,u;\theta)$ denote ith order derivatives of f,g with respect to t. Note that $\frac{\mathrm{d}^i}{\mathrm{d}t^i}f(x,u;\theta)$, $\frac{\mathrm{d}^i}{\mathrm{d}t^i}g(x,u;\theta)$ can be regarded as polynomials of $\frac{\mathrm{d}^i x}{\mathrm{d}t^i},\dots,x,\frac{\mathrm{d}^i u}{\mathrm{d}t^i},\dots,u$. For a non-negative integer i, let

$$f^{(i)}(x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)}),\ g^{(i)}(x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)})\in\mathbb{C}(\theta)[x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)}] \tag{15}$$

denote the polynomials obtained by replacing $\frac{\mathrm{d}^i x}{\mathrm{d}t^i},\dots,x,\frac{\mathrm{d}^i u}{\mathrm{d}t^i},\dots,u$ of $\frac{\mathrm{d}^i}{\mathrm{d}t^i}f(x,u;\theta)$, $\frac{\mathrm{d}^i}{\mathrm{d}t^i}g(x,u;\theta)$ with $x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)}$, respectively. Thus, using the notations in Table 1, (14a), (14b) can be represented as polynomial equations

$$\begin{pmatrix}x^{(1)}\\ x^{(2)}\\ \vdots\\ x^{(N+1)}\end{pmatrix}=\begin{pmatrix}f^{(0)}(x^{(0)},u^{(0)};\theta)\\ f^{(1)}(x^{(1)},x^{(0)},u^{(1)},u^{(0)};\theta)\\ \vdots\\ f^{(N)}(x^{(N)},\dots,x^{(0)},u^{(N)},\dots,u^{(0)};\theta)\end{pmatrix}, \tag{16a}$$

$$\begin{pmatrix}y^{(0)}\\ y^{(1)}\\ \vdots\\ y^{(N)}\end{pmatrix}=\begin{pmatrix}g^{(0)}(x^{(0)},u^{(0)};\theta)\\ g^{(1)}(x^{(1)},x^{(0)},u^{(1)},u^{(0)};\theta)\\ \vdots\\ g^{(N)}(x^{(N)},\dots,x^{(0)},u^{(N)},\dots,u^{(0)};\theta)\end{pmatrix}, \tag{16b}$$

where $\frac{\mathrm{d}^i}{\mathrm{d}t^i}f,\frac{\mathrm{d}^i}{\mathrm{d}t^i}g$ are replaced with $f^{(i)},g^{(i)}$. To consider the order of derivatives required to extract the input–output relations in (12a), (12b), let us denote a set of polynomials obtained through algebraic manipulations of the model differentiated up to the mth order as $I^{(m)}$. Specifically, for (12a), (12b), given a non-negative integer m, we define the ideal $I^{(m)}$ as follows:

$$\begin{aligned}I^{(m)}=&\langle x^{(1)}-f^{(0)},\dots,x^{(m)}-f^{(m-1)},\dots,y^{(0)}-g^{(0)},\dots,y^{(m)}-g^{(m)}\rangle\\ &\subset\mathbb{C}(\theta)[x^{(m)},\dots,x^{(0)},y^{(m)},\dots,y^{(0)},u^{(m)},\dots,u^{(0)}],\end{aligned} \tag{17}$$

where $f^{(i)}(x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)};\theta),g^{(i)}(x^{(i)},\dots,x^{(0)},u^{(i)},\dots,u^{(0)};\theta)$ are denoted as $f^{(i)},g^{(i)}$, for $i=0,\dots,m$, respectively. Using (17), a set of polynomials in which $x^{(m)},\dots,x^{(0)}$ are eliminated are described as follows:

$$I^{(m)}\cap\mathbb{C}(\theta)[y^{(m)},\dots,y^{(0)},u^{(m)},\dots,u^{(0)}],$$

which is a set of polynomials representing the input–output relationships of the model (Fig. 1d).

Proposition 1 shows the existence of the equations representing the input–output relations of the physical computing systems modeled

by (12a), (12b). More precisely, it is shown that (19) has a non-zero polynomial, and also the specific number of derivations needed for the existence is provided.

Proposition 1 (*Theorem Shown in [28]*). *For a model (12a), (12b) with $f(x,u;\theta), g(x,u;\theta) \in \mathbb{C}(\theta)[x,u]$, let us consider $I^{(N)}$ where N is the dimension of the state vector of the system:*

$$I^{(N)} = \langle x^{(1)} - f^{(0)}, \dots, x^{(N)} - f^{(N-1)}, \dots, y^{(0)} - g^{(0)}, \dots, y^{(N)} - g^{(N)} \rangle \subset \mathbb{C}(\theta)[x^{(N)}, \dots, x^{(0)}, y^{(m)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]. \quad (18)$$

Then,

$$I^{(N)} \cap \mathbb{C}(\theta)[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}], \quad (19)$$

is not zero ideal.

The following Lemma shown in [28] is used to prove Proposition 1.

Lemma 1 (*Theorem 1.6 of [28]*). *Let K be the field. Polynomials $p_1, \dots, p_S \in K[x_1, \dots, x_s]$ are algebraically dependent over K if $S > s$, that is, there exists a non-zero polynomial $P \in k[z_0, \dots, z_S]$ such that $P(p_1, \dots, p_S) = 0$.*

Proof of Proposition 1. Regarding $y^{(N)}, \dots, y$ satisfying (12a), (12b) and their higher order derivatives up to the Nth order, the following holds:

$$y^{(N)}, \dots, y^{(0)} \in \mathbb{C}(u^{(N)}, \dots, u^{(0)}, \theta)[x_1^{(0)}, \dots, x_N^{(0)}], \quad (20)$$

because the higher order derivatives of $x_1, \dots, x_N$ of (12a) can be represented by those with the lower order derivatives. According to Lemma 1, $y^{(i)} (i = 0, \dots, N)$ being algebraically dependent over $\mathbb{C}(u^{(N)}, \dots, u^{(0)}, \theta)$ holds. Thus, a non-zero polynomial $P \in \mathbb{C}(u^{(N)}, \dots, u^{(0)}, \theta)[z_0, \dots, z_N]$ exists such that $P(y^{(N)}, \dots, y^{(0)}) = 0$. By clearing the denominators of the coefficients of $P(y^{(N)}, \dots, y^{(0)})$, we obtain a polynomial in $I^{(N)} \cap \mathbb{C}(\theta)[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]$. This concludes the proof of Proposition 1. □

Next, we explain how to characterize (19), which represents the input–output relations of the model. In terms of algebra, this problem involves describing the ideal where the Gröbner basis is known to be a key.

Definition 9 (*Gröbner Basis*). For an ideal I on a polynomial ring $K[x]$ with a given monomial order, if its generator $G = \{g_1, \dots, g_s\}$ satisfies

$$p \in I \Leftrightarrow p \text{ is divisible by } G, \quad (21)$$

then G is called the Gröbner basis of I.

For an arbitrary non-zero ideal on polynomial rings with a fixed monomial order, there exists a Gröbner basis [25]. Although the Gröbner basis of an ideal with respect to a fixed monomial order is not unique, the following is unique.

Definition 10 (*Reduced Gröbner Basis*). For an ideal I on a polynomial ring $K[x]$ with a given monomial order, if $G = \{g_1, \dots, g_s\}$ is the Gröbner basis of I and satisfies the followings:

- $\mathrm{LC}(g_i)$ is 1 for all $i = 1, \dots, s$; and
- if $i \neq j$, any monomial in the set of monomials, whose coefficients are non-zero, appear in g_i is not divisible by $\mathrm{LM}(g_j)$,

then G is the reduced Gröbner basis of I.

According to (21), any polynomial in an ideal can be represented as a linear combination of its Gröbner basis. Furthermore, according to (21), the Gröbner basis is known to solve a membership problem, i.e., discriminate whether a given polynomial is in the ideal [25], and thus characterize the ideal. Therefore, (19) can be characterized by its Gröbner basis. In particular, any polynomial equation corresponding to a polynomial in (19) holds if all the equations constituting the Gröbner basis for (19) hold. In other words, the Gröbner basis for (19) explicitly describes the input–output relations for a given physical computing system. Notably, the Gröbner basis can be derived automatically using computer algebra software. Furthermore, there is another benefit of using the Gröbner basis. The Gröbner basis can be used to determine whether a given differential equation belongs to (19). This allows us to compare different physical implementations of a physical computing system, and thus, solve Target task 2 as explained in Section 4.3.

A type of lexcographic order shown in the following examples is what we used to compute the Gröbner basis and derive input–output equations.

Example 1. For a physical computing system modeled as (12a), (12b), we consider the polynomial ring

$$\mathbb{C}(\theta)[x^{(N)}, \dots, x^{(0)}, y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}] \quad (22)$$

as in Proposition 1. To eliminate $x^{(N)}, \dots, x^{(0)}$ from the ideal (18) corresponding to the model, we fix the monomial order of the ring to the lexcographic order such that $x^{(N)} > \cdots > x^{(0)} > y^{(N)} > \cdots > y^{(0)} > u^{(N)} > \cdots > u^{(0)}$. That is, monomials in the ring are considered as $\left(x^{(N)}\right)^{\alpha_1} \left(x^{(N-1)}\right)^{\alpha_2} \cdots \left(u^{(1)}\right)^{\alpha_{s-1}} \left(u^{(0)}\right)^{\alpha_s}$, where the derivatives (e.g., $x^{(N)}$) are considered distinct from the ordinary variables (e.g., x).

In Section 4.3, we consider the polynomial ring $\mathbb{C}(\theta)[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]$, which is the subset of the ring considered in Example 1. The input–output order defined in Definition 11 is the order induced by the lexcographic order explained in Example 1. That is, we fix the monomial order of the ring to the lexcographic order such that $y^{(N)} > \cdots > y^{(0)} > u^{(N)} > \cdots > u^{(0)}$. This corresponds to considering monomials in the ring as $\left(y^{(N)}\right)^{\beta_1} \left(y^{(N-1)}\right)^{\beta_2} \cdots \left(u^{(1)}\right)^{\beta_{s-1}} \left(u^{(0)}\right)^{\beta_s}$.

Definition 11 (*Input–Output Order*). Let us denote the ring $\mathbb{C}(\theta)[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]$ as $\mathbb{C}(\theta)[z]$, in which each element of $z = (z_1, z_2, \dots, z_s)$ corresponds to $y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}$ in order. Let us consider a monomial order $<$ such that for arbitrary $q = z_1^{\alpha_1} z_2^{\alpha_2}, \dots, z_s^{\alpha_s}, p = z_1^{\beta_1} z_2^{\beta_2}, \dots, z_s^{\beta_s} \in \mathbb{C}(\theta)[z]$, $q < p$ holds if the left-most element among the non-zero elements of $(\beta_1 - \alpha_1, \beta_2 - \alpha_2, \dots, \beta_s - \alpha_s)$ is positive. We define $<$ as an input–output order.

An element of the Gröbner basis of (19), representing the input–output relationships of a model of physical computing system (12a), (12b). It is uniquely obtained by fixing the monomial order to the one appears in Example 1 and computing the reduced Gröbner basis. For simplicity, we consider the polynomial equation obtained in this way, which is unique, and denote it as follows:

$$F(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \theta) = 0, \quad (23)$$

where $F(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \theta) \in \mathbb{C}(\theta)[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]$. The input–output equation of (12a), (12b) can be obtained by replacing $y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}$ of (23) with $\frac{\mathrm{d}^N y}{\mathrm{d}t^N}, \dots, y, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u$ as follows:

$$F\left(\frac{\mathrm{d}^N y}{\mathrm{d}t^N}, \dots, y, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u; \theta\right) = 0. \quad (24)$$

4.2. Main Theorem 1: Deriving design guidelines of physical computing systems algebraically

Firstly, we consider Target Task 1 (time-series generations.) Theorem 1 shown below is a fundamental theorem for deriving design guidelines from (24) for Target Task 1. Intuitively, it provides a discriminator to determine whether a given physical computing system generates $\bar{y}(t)$, which reveals design guidelines for the system.

We first introduce some definitions. Let $\mathcal{F}_y$ denote a set of sufficiently smooth functions containing functions from $\mathbb{R}$ to $\mathbb{R}$.

Definition 12 (*Consistent Initial Values*)**.** For a given input $u \in \mathcal{F}_u$ and initial values of (12a), denoted as $x(0) \in \mathbb{R}^N$, we define the consistent initial values of (14a) as $\left(x(0), \dots, \frac{\mathrm{d}^N x}{\mathrm{d}t^N}\Big|_{t=0}\right) \in \mathbb{R}^{N\times(N+1)}$ that satisfies

$$\begin{pmatrix} \frac{\mathrm{d}x}{\mathrm{d}t}\Big|_{t=0} \\ \vdots \\ \frac{\mathrm{d}^N x}{\mathrm{d}t^N}\Big|_{t=0} \end{pmatrix} = \begin{pmatrix} f^{(0)}(x(0), u(0); \theta) \\ \vdots \\ f^{(N-1)}\left(\frac{\mathrm{d}^{N-1}x}{\mathrm{d}t^{N-1}}\Big|_{t=0}, \dots, x(0), \frac{\mathrm{d}^N u}{\mathrm{d}t^N}\Big|_{t=0}, \dots, u(0); \theta\right) \end{pmatrix}. \tag{25}$$

According to (15), $f^{(i)}$ is a function of $x^{(0)}, \dots, x^{(i)}, u^{(0)}, \dots, u^{(i)}$ for $i = 0, \dots, N-1$. Hence, for arbitrary $x(0) \in \mathbb{R}^N, u \in \mathcal{F}_u$, by substituting $\left(x(0), \dots, \frac{\mathrm{d}^i x}{\mathrm{d}t^i}\Big|_{t=0}, u(0), \dots, \frac{\mathrm{d}^i u}{\mathrm{d}t^i}\Big|_{t=0}\right)$ into $\left(x^{(0)}, \dots, x^{(i)}, u^{(0)}, \dots, u^{(i)}\right)$ of $f^{(i)}$ in the order of $i = 0, \dots, N-1$, consistent initial values $\left(x(0), \dots, \frac{\mathrm{d}^N x}{\mathrm{d}t^N}\Big|_{t=0}\right)$ can be obtained.

Lemma 2. *Suppose that the system of differential Eqs. (12a) with a given $u \in \mathcal{F}_u$ and a given initial state vector $x(0)$ admits a unique solution. Then, the system of differential equations that consists of (12a) and its higher order derivatives up to the Nth order (14a) admits a unique solution given consistent initial values. Furthermore, the output satisfying the observation Eq. (12b) and their derivatives up to the Nth order are also unique given the unique solution of (14a) .*

Proof. Let us substitute the unique solution of (12a) into their right side, $f(x, u; \theta)$. We then obtain a unique $\frac{\mathrm{d}x}{\mathrm{d}t}$, which is well defined because $f(x, u; \theta) = f^{(0)}(x, u; \theta)$ is a polynomial vector with respect to x and u. Then, $\frac{\mathrm{d}x}{\mathrm{d}t}\Big|_{t=0} = f(x(0), u(0); \theta) = f^{(0)}(x, u; \theta)$ holds. By substituting $x(t), \frac{\mathrm{d}x}{\mathrm{d}t}, u(t), \frac{\mathrm{d}u}{\mathrm{d}t}$ into the right side of the derivative of the system of differential equations, $\frac{\mathrm{d}}{\mathrm{d}t} f(x, u; \theta)$, we obtain a unique $\frac{\mathrm{d}^2 x}{\mathrm{d}t^2}$. $\frac{\mathrm{d}^2 x}{\mathrm{d}t^2}$ is well defined because $\frac{\mathrm{d}}{\mathrm{d}t} f(x, u; \theta) = f^{(1)}(\frac{\mathrm{d}x}{\mathrm{d}t}, x, \frac{\mathrm{d}u}{\mathrm{d}t}, u; \theta)$ is a polynomial vector with respect to $\frac{\mathrm{d}x}{\mathrm{d}t}, x, \frac{\mathrm{d}u}{\mathrm{d}t}, u$. Thus, it holds that $\frac{\mathrm{d}^2 x}{\mathrm{d}t^2}\Big|_{t=0} = f^{(1)}(\frac{\mathrm{d}x}{\mathrm{d}t}\Big|_{t=0}, x(0), \frac{\mathrm{d}u}{\mathrm{d}t}\Big|_{t=0}, u(0); \theta)$. By repeating these procedures, we obtain a unique $x(t), \frac{\mathrm{d}x}{\mathrm{d}t}, \dots, \frac{\mathrm{d}x^N}{\mathrm{d}t^N}$, that is, the solution of (14a) given consistent initial values (25). Furthermore, $y(t), \frac{\mathrm{d}y}{\mathrm{d}t}, \dots, \frac{\mathrm{d}^N y}{\mathrm{d}t^N}$ are shown to be unique by substituting $x(t), \frac{\mathrm{d}x}{\mathrm{d}t}, \dots, \frac{\mathrm{d}^N x}{\mathrm{d}t^N}$ into the right side of (14b),

$$g(x, u; \theta) = g^{(0)}(x, u), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N} g(x, u; \theta) = g^{(N)}\left(\frac{\mathrm{d}^N x}{\mathrm{d}t^N}, \dots, x, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u; \theta\right). \quad \square \tag{26}$$

Definition 13 (*Consistent Outputs*)**.** Suppose that the system of differential Eqs. (12a) with a given $u \in \mathcal{F}_u$ and a given initial state vector admits a unique solution. We define the consistent outputs of (14a), (14b) as $y, \dots, \frac{\mathrm{d}^N y}{\mathrm{d}t^N}$ such that the unique solution of (14a) given consistent initial values are substituted into the right-side of (14b).

Assumption 3 (*Generic Physical Computing System*)**.** Given a physical computing system (12a), (12b) and initial values of the consistent outputs of (14a), (14b), $(y(0), \dots, \frac{\mathrm{d}^N y}{\mathrm{d}t^N}\Big|_{t=0}) \in \mathbb{R}^{N+1}$, we assume that the input–output Eq. (24) given initial values $(y(0), \dots, \frac{\mathrm{d}^{N-1} y}{\mathrm{d}t^{N-1}}\Big|_{t=0}) \in \mathbb{R}^N$ admits a unique solution for arbitrary $u \in \mathcal{F}_u$.

Assumption 4 (*Generic Parameter Vectors*)**.** For a physical computing system (12a), (12b) with the input–output Eq. (24), a specific parameter vector $\theta_* \in \mathbb{R}^n$ is defined as generic if the input–output equation of (12a), (12b) with $\theta = \theta_*$ is equivalent to (24) with θ_* substituted for θ. Throughout this work, we consider generic parameter vectors and a set of generic parameter vectors $\Theta \subset \mathbb{R}^n$.

In general, generic parameters can be configured using a comprehensive Gröbner system [29–31]. See Appendix B for additional comments on generic parameter vectors.

Theorem 1. *Let $\Theta \subset \mathbb{R}^n$ be a set of generic parameter vectors and $\bar{y}(t) \in \mathcal{F}_y$ be a target time-series. Suppose that a physical computing system (12a) , (12b) satisfies Assumption 3. Let $\theta \in \Theta, u \in \mathcal{F}_u, x(0) \in \mathbb{R}^N$ be given and fixed. Let us denote the consistent outputs of (14a), (14b) as $\hat{y}(t; \theta, u, x(0)), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t; \theta, u, x(0))$. Then, $\bar{y}(t)$ can be generated by (12a), (12b) given $x(0) \in \mathbb{R}^N$ on a time interval $[0, T]$ if and only if*

$$F\left(\frac{\mathrm{d}^N \bar{y}}{\mathrm{d}t^N}, \dots, \bar{y}(t), \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u(t); \theta\right) = 0, \tag{27}$$

holds over $[0, T]$ and

$$\hat{y}(0; \theta, u, x(0)) = \bar{y}(0), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t; \theta, u, x(0))\Big|_{t=0} = \frac{\mathrm{d}^N \bar{y}}{\mathrm{d}t^N}\Big|_{t=0}. \tag{28}$$

Proof of Theorem 1. Let us take an arbitrary $(\theta, u, x(0)) \in \Theta \times \mathcal{F}_u \times \mathbb{R}^N$. According to Lemma 2, $\hat{y}(t; \theta, u, x(0)), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t; \theta, u, x(0))$ is unique.

First, we show that for any $x(0) \in \mathbb{R}^N$, $\hat{y}(t; \theta, u, x(0)) = \bar{y}(t)$ holds over $[0, T]$ if (27) and (28) hold. Considering that F is in (18) and $\theta \in \Theta$, it follows that

$$F\left(\frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t; \theta, u, x(0)), \dots, \hat{y}(t; \theta, u, x(0)), \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u(t); \theta\right) = 0.$$

Considering that (12a), (12b) is generic and that both $\hat{y}(t; \theta, u, x(0))$ and $\bar{y}(t)$ satisfy (24), given the initial condition (28), $\hat{y}(t; \theta, u, x(0))$ and $\bar{y}(t)$ must coincide over $[0, T]$.

Next, we show that for any $x(0) \in \mathbb{R}^N$, (27) and (28) holds if $\bar{y}(t)$ can be generated by (12a), (12b). Let us again take an arbitrary $x(0) \in \mathbb{R}^N$. Because $\bar{y}(t) \in \mathcal{F}_y$ can be generated by (12a), (12b), this is equivalent to $\hat{y}(t; \theta, u, x(0))$ up to the Nth order, where $\hat{y}(t; \theta, u, x(0))$ is a unique solution of (12a), (12b), as shown in Lemma 2. Hence, it holds that (28). Furthermore, $\left(\hat{y}(t; \theta, u, x(0)), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t; \theta, u, x(0))\right)$ satisfies (24) because F is in $I^{(N)}$ and $\theta \in \Theta$. Therefore, $\left(\bar{y}, \dots, \frac{\mathrm{d}^N \bar{y}}{\mathrm{d}t^N}\right)$ satisfies F, that is, (27) holds. This concludes the proof. $\square$

Theorem 1 guarantees that (24) plays a role in deriving the design guidelines of the system for the time-series generation task. A "specific design guideline" for Target task 1 is the one explains how to adjust a system to generate a target time series. In existing literatures [15,20], for physical computing systems that are static feedback linearizable, specific design guidelines in the form of feedback control laws have been derived. On the other hand, our approach allows more general physical computing systems, described as (12a), (12b) and obtains the specific design guideline (27). (27) indicates that if inputs satisfying (27) are applied to the system, the target time series can be generated. In this sense, (27) serves as a specific design guideline for physical computing systems if (28) holds. It is noteworthy that our approach derives (27) explicitly and systematically based on algebraic theory and Gröbner bases computation. (28) is critical for non-stationary tasks, whereas it is not for stationary ones. See Experiment 3 for examples.

Experiment 1. *Designing inputs applied to physical computing systems.* We consider the physical system modeled using the FitzHugh–Nagumo equation [32],

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_1(x_1 - 1)\left(1 - \frac{1}{3}x_1\right) - x_2 + u, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_1. \tag{29}$$

We assume that x_2 is observed; $y = x_2$. Then, following the procedures explained in Example A.2 in Supplementary material, the input–output equation of this system, $y + \frac{1}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^3 - \frac{4}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + \frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} - u = 0$ is obtained.

According to Theorem 1, the input–output equation in which a target time-series is substituted, provides specific design guidelines for generation tasks. Manipulating $y + \frac{1}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^3 - \frac{4}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + \frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} - u = 0$, a design guideline

$$u = \bar{y} + \frac{1}{3}\left(\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\right)^3 - \frac{4}{3}\left(\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\right)^2 + \frac{\mathrm{d}\bar{y}}{\mathrm{d}t} + \frac{\mathrm{d}^2 \bar{y}}{\mathrm{d}t^2}, \tag{30}$$

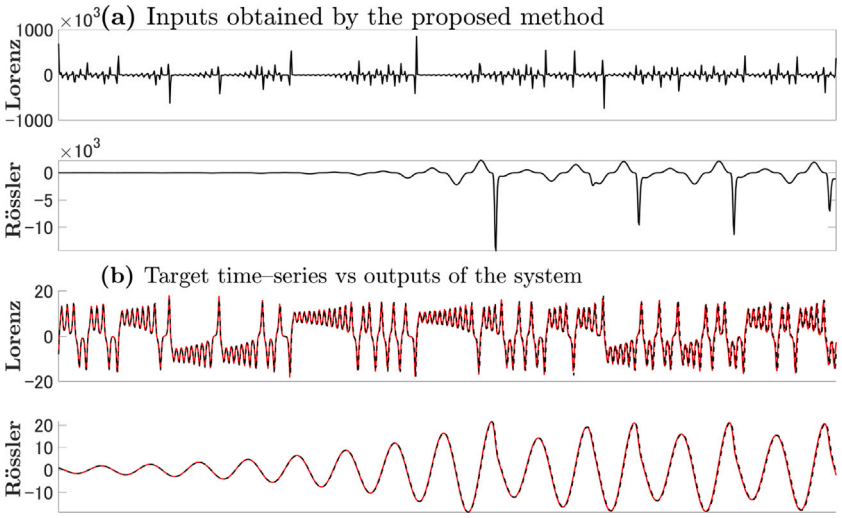

Fig. 2. Generation of chaotic time-series (the Lorentz and the Rössler equations) by the physical computing system (29).

is derived. Let us set an initial states of (29) as $(x_1(0), x_2(0))^\top = (\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\big|_{t=0}, \bar{y}(0))^\top$. We obtain $(\frac{\mathrm{d}x_1}{\mathrm{d}t}\big|_{t=0}, \frac{\mathrm{d}x_2}{\mathrm{d}t}\big|_{t=0})^\top$ by substituting $(x_1(0), x_2(0))^\top$ into the right-side of top 2 equations of (25). By substituting $(x_1(0), x_2(0), \frac{\mathrm{d}x_1}{\mathrm{d}t}\big|_{t=0}, \frac{\mathrm{d}x_2}{\mathrm{d}t}\big|_{t=0})^\top$ into the third and fourth equations, we obtain $(\frac{\mathrm{d}^2x_1}{\mathrm{d}t^2}\big|_{t=0}, \frac{\mathrm{d}^2x_2}{\mathrm{d}t^2}\big|_{t=0})^\top$. By repeating these procedures, we obtain a consistent initial values. Given the consistent initial values, the outputs of (29) and their derivatives up to 2nd order, $\hat{y}(t;\theta,u,x(0))$, $\frac{\mathrm{d}}{\mathrm{d}t}\hat{y}(t;\theta,u,x(0))$, $\frac{\mathrm{d}^2}{\mathrm{d}t^2}\hat{y}(t;\theta,u,x(0))$ consistent. Since it holds that $y = x_2$ and $\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_1$, it must hold that $\hat{y}(0) = x_2(0) = \bar{y}(0)$ and $\frac{\mathrm{d}}{\mathrm{d}t}\hat{y}(t;\theta,u,x(0))\big|_{t=0} = x_1(0) = \frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\big|_{t=0}$, which satisfies (28). Using the input defined as (30) and given initial values $(x_1(0), x_2(0))^\top = (\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\big|_{t=0}, \bar{y}(0))^\top$, the system generates the given time-series, $\bar{y}$. The guideline derived based on Theorem 1, (30), suggests that the system can generate several target time-series by adjusting its input.

We now numerically confirm how designed inputs based on Theorem 1 generate target time-series well. We consider target dynamics generated by chaotic systems called the Lorentz system [33],

$$\frac{\mathrm{d}\bar{x}_1}{\mathrm{d}t} = 10(\bar{x}_2 - \bar{x}_1), \quad \frac{\mathrm{d}\bar{x}_2}{\mathrm{d}t} = \bar{x}_1(28 - \bar{x}_3) - \bar{x}_2, \quad \frac{\mathrm{d}\bar{x}_3}{\mathrm{d}t} = \bar{x}_1\bar{x}_2 - \frac{8}{3}\bar{x}_3, \tag{31}$$

and the Rössler system [34],

$$\frac{\mathrm{d}\bar{x}_1}{\mathrm{d}t} = -\bar{x}_2 - \bar{x}_3, \quad \frac{\mathrm{d}\bar{x}_2}{\mathrm{d}t} = \bar{x}_1 + \frac{1}{10}\bar{x}_2, \quad \frac{\mathrm{d}\bar{x}_3}{\mathrm{d}t} = \frac{1}{10} + \bar{x}_3(\bar{x}_1 - 14). \tag{32}$$

In this experiment, $\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\big|_{t=0} = x_1(0)$ are estimated via the approximation of the numerical solution of $\bar{y}$ of (31) and (32) using polynomial interpolants in the Chebyshev points [35]. We assume that data points are sampled from the first variables of each system. We substituted the discretized target time-series into the design guideline (30) to obtain inputs. Then, they are applied to (29). Fig. 2 shows the designed inputs and the outputs of the physical computing system for generating the two types of time-series. Thus, the different time-series, both of which are generated by chaotic dynamics, are numerically confirmed to be generated by the physical computing system based on the design guidelines derived by our approach. Notably, this property, which is a type of multi-tasking property, is desirable in physical computing, leading to the efficient use of given physical assets to perform various computations.

As a conclusion of this experiment, the relationship between our approach and differential flatness [21–24] is commented. In general, (12a) is differentially flat if there exist relations

$$\xi : \mathbb{R}^N \times \left(\mathbb{R}^M\right)^{r+1} \to \mathbb{R}^M,$$
$$\phi : \left(\mathbb{R}^M\right)^{r} \to \mathbb{R}^N,$$
$$\psi : \left(\mathbb{R}^M\right)^{r+1} \to \mathbb{R}^M$$

such that

$$\begin{aligned} y &= \xi\left(x, u, \frac{\mathrm{d}u}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}^r u}{\mathrm{d}t^r}\right), \\ x &= \phi\left(y, \frac{\mathrm{d}y}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}^r y}{\mathrm{d}t^r}\right), \\ u &= \psi\left(y, \frac{\mathrm{d}y}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}^r y}{\mathrm{d}t^r}\right). \end{aligned} \tag{33}$$

The output y of (33) is called the flat output. The physical computing system considered in this experiment is represented as (29) with $\xi(x) = x_2$. Then, it is easy to find that $\phi(y, \frac{\mathrm{d}y}{\mathrm{d}t})$ can be taken as $(\frac{\mathrm{d}y}{\mathrm{d}t}, y)^\top$. In addition, the specific design guideline obtained through the proposed approach, (30) shows that it holds that $\psi(y, \frac{\mathrm{d}y}{\mathrm{d}t}, \frac{\mathrm{d}^2y}{\mathrm{d}t^2}) = y + \frac{1}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^3 - \frac{4}{3}\left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + \frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\mathrm{d}^2y}{\mathrm{d}t^2}$. These conclude that the physical computing system considered in Experiment 1 is differentially flat. Note that this conclusion is obtained systematically based on the algebraic framework based on Gröbner basis computation. In general, the differential flatness property enables the transformation of the original system into a linear canonical form [36]. Subsequently, control methods for linear systems, which are well-developed, become applicable in designing inputs for the original system. Control methods based on such properties is called flatness-based controls [24] and one of the examples is provided below. The physical computing system represented by (29) and $y = x_2$ can be transformed into the linear system

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + v, \quad (y_1, y_2)^\top = \left(y, \frac{\mathrm{d}y}{\mathrm{d}t}\right)^\top, \tag{34}$$

where $v : \mathbb{R} \to \mathbb{R}$ denotes the input function satisfying

$$u = y_1 + \frac{1}{3}{y_2}^3 - \frac{4}{3}{y_2}^2 + y_2 + v. \tag{35}$$

Note that (35) is obtained based on our design guideline (30). In the linear system (34), v can be designed so that $(y_1, y_2)^\top$ fits to $(\bar{y}, \frac{\mathrm{d}\bar{y}}{\mathrm{d}t})^\top$ by control methods such as a state feedback control based on pole placement techniques [37]. Then, by substituting v together with y_1, y_2 into (35), the input to the physical computing system u is obtained. It is also noteworthy that designing u based on (35) does not require the second order derivatives $\frac{\mathrm{d}^2\bar{y}}{\mathrm{d}t^2}$ unlike designing u based on (30) solely. In summary, our proposed design guideline (30) is shown to be applicable for Target task 1, and moreover, incorporation of the concept of flatness into it is promising, which may enable more practical use of physical

computing systems. We leave further applications of flatness-based controls as future work.

4.3. Main Theorem 2: Relating two different physical computing systems

In this section, we provide a theorem that enables the design of physical computing systems that can be replaced with other computing systems. Such a design is useful considering the diverse implementations of physical systems (Target task 2.) The basic idea is to provide a description of the relationship between the input–output equations of a given physical computing system and a target system. Such a relationship provides information about whether those systems can be interchanged, as well as providing design guidelines.

To accomplish this, we leverage another benefit of the Gröbner basis. The Gröbner basis (e.g., (24)) can determine whether a given polynomial in a set of polynomials corresponds to itself (e.g., (19)); a polynomial is shown to be or not to be in a certain ideal through division by its Gröbner basis. Suppose that the input–output equation of a target system, which is denoted as h in Assumption 2, is set as the polynomial to be divided. Then, intuitively, the division by (24) makes it possible to determine whether the target input–output behavior is realized by a given physical computing system. The monomial order that we introduce for this purpose is the input–output order, defined in Definition 11 The following is a example of the input–output order.

Example 2 (*The Input–Output Order*). Let us consider a polynomial, $h := y^{(2)} + y^{(0)} - \left(1 - \left(y^{(0)}\right)^2\right) y^{(1)}$ where the order of variables is $y^{(2)}, y^{(1)}, y^{(0)}$, as discussed in Experiment 3. The monomials in h are $y^{(2)}, y^{(0)}, y^{(1)}$, and $y^{(1)}\left(y^{(0)}\right)^2$. Given the input–output order, they are ordered as $y^{(2)} > y^{(1)}\left(y^{(0)}\right)^2 > y^{(1)} > y^{(0)}$.

The monomial with the highest order in polynomial p with respect to order $<$ is denoted as $\mathrm{LM}_<(p)$ and the coefficient of $\mathrm{LM}_<(p)$ is denoted as $\mathrm{LC}_<(p)$ (See Definitions 6 and 7). For instance, consider h appears in Example 2. $\mathrm{LM}_<(h) = y^{(2)}$ and $\mathrm{LC}_<(h) = 1$ hold with respect to the input–output order. With these notations, a theorem on the replaceability is provided. Let us consider the correspondence between $h = 0$ and the input–output equation of a target system for the practical use of the theorem.

Theorem 2. *Given and fixed $u \in \mathcal{F}_u$ and $\tilde{\theta} \in \mathbb{R}^{\tilde{n}}$, suppose that there exists a target time-series $\bar{y}(t) \in \mathcal{F}_y$ satisfying a differential equation $h\left(\frac{\mathrm{d}^N\bar{y}}{\mathrm{d}t^N}, \dots, \bar{y}, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u; \tilde{\theta}\right) = 0$ defined over $[0,T]$ and Assumptions 1 and 2. Let us consider a physical computing system described by (12a), (12b) satisfies Assumption 3. Let $\Theta \subset \mathbb{R}^n$ be a set of generic parameters. Let $\theta \in \Theta$ be given and fixed. We fix a monomial order of $\mathbb{C}(\theta,\tilde{\theta})[y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}]$ as the input–output order $<$. Suppose that $h(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \tilde{\theta})$ is divided by F that is defined in (24), and represented as*

$$h(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \tilde{\theta}) = q(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \theta, \tilde{\theta})F + R(y^{(N)}, \dots, y^{(0)}, u^{(N)}, \dots, u^{(0)}; \theta, \tilde{\theta}), \tag{36}$$

where we assume that $\mathrm{LM}_<(h) \geq \mathrm{LM}_<(F)$, $\mathrm{LC}_<(F)$ is non-zero, and the coefficient of h with respect to $\mathrm{LM}_<(F)$ is non-zero. Let us take arbitrary initial values $x(0) \in \mathbb{R}^N$ of (12a) and denote the consistent initial values of (14a) as $(x(0), \dots, \frac{\mathrm{d}^N x}{\mathrm{d}t^N}\big|_{t=0}) \in \mathbb{R}^{N\times(N+1)}$. Let us denote the consistent outputs of (14a), (14b) as $\hat{y}(t;\theta,u,x(0)), \dots, \frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t;\theta,u,x(0))$. Then, $\bar{y}(t)$ can be generated by (12a), (12b) given $x(0)$ if

$$R\left(\frac{\mathrm{d}^N\bar{y}}{\mathrm{d}t^N}, \dots, \bar{y}(t), \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u(t); \theta, \tilde{\theta}\right) = 0 \tag{37}$$

holds over $[0,T]$ and (28) holds.

Remark 1. To apply Theorem 2 to Target task 2, one may consider that $h\left(\frac{\mathrm{d}^N y}{\mathrm{d}t^N}, \dots, y, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u; \tilde{\theta}\right) = 0$ is the input–output equation of a system, which is regarded as a Target system, of which parameters vector is denoted as $\tilde{\theta}$.

Remark 2. The basic assumptions are the same as those for Theorem 1 together with Assumption 2. Additional assumptions are related to the order of the monomials. In general, when the polynomials are divided, the polynomials to be divided have higher-order monomials than those used for the division. Hence, to consistently perform the division, it must hold that $\mathrm{LM}_<(h) \geq \mathrm{LM}_<(F)$, which is one of the additional assumptions. In addition, we assume that $\mathrm{LC}_<(F)$ and $\mathrm{LC}_<(h)$ are non-zero or non-zero functions over $[0,T]$, which are generic ones.

Proof. Because $\mathrm{LM}_<(h) \geq \mathrm{LM}_<(F)$ holds, q in (36) is represented by $q'/\mathrm{LC}(F)$, that is, $q = q'/\mathrm{LC}(F)$. According to the assumptions, $\mathrm{LC}_<(F)$ is non-zero and q' is non-zero; thus, q is non-zero. By substituting $(\bar{y}, \dots, \frac{\mathrm{d}^N\bar{y}}{\mathrm{d}t^N})$ into $(y, \dots, \frac{\mathrm{d}^N y}{\mathrm{d}t^N})$ of (36), the left side of (36) and R take zeros over $t \in [0,T]$ because $\bar{y}$ is a solution of

$$h\left(\frac{\mathrm{d}^N y}{\mathrm{d}t^N}, \dots, y, \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u; \tilde{\theta}\right) = 0$$

and (37). Thus, it must hold that

$$F\left(\frac{\mathrm{d}^N\bar{y}}{\mathrm{d}t^N}, \dots, \bar{y}(t), \frac{\mathrm{d}^N u}{\mathrm{d}t^N}, \dots, u(t); \theta\right) = 0 \tag{38}$$

over $t \in [0,T]$. This is equivalent to (27). Hence, by connecting this to the sufficiency of Theorem 1, we finish the proof. □

(37) is a differential polynomial equation up to the Nth order, whose coefficients are functions of $\theta, \tilde{\theta}$ and the variables are u. In the same manner as for (27), design guidelines for a physical computing system can be derived from (37). However, in this case, they are related to the replaceability of (12a), (12b) with a target system generating $\bar{y}$.

Experiment 3. *Exploitation of the replaceability and subsequent design.* We apply Theorem 2 and investigate specific design guidelines for a physical computing system to be utilized to replace a target system that generates parameterized time-series. Let us consider a physical computing system modeled by a FitzHugh–Nagumo equation with parameters $(\theta_1, \dots, \theta_4)^\top \in \mathbb{R}^4$,

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \theta_1\left(x_1 - \frac{1}{3}x_1^3 + x_2 + \theta_2\right), \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = -\frac{1}{\theta_1}\left(x_1 - \theta_3 + \theta_4 x_2\right) \tag{39}$$

with readout $y = x_1$. We denote this system as Σ_1 and regard this as the system to be designed. The input–output equation of Σ_1 is as follows:

$$\theta_4 y^3 + 3\theta_1\frac{\mathrm{d}y}{\mathrm{d}t}y^2 + 3\left(1-\theta_4\right)y + 3\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} - 3\frac{\mathrm{d}y}{\mathrm{d}t}\left(\theta_1 - \frac{\theta_4}{\theta_1}\right) - 3\left(\theta_3 - \theta_2\theta_4\right) = 0. \tag{40}$$

We denote the left side of (40) as F_1. Then, we consider a target system that consists of van der Pol equation [38], $\frac{\mathrm{d}\tilde{x}_1}{\mathrm{d}t} = \tilde{x}_2$, $\frac{\mathrm{d}\tilde{x}_2}{\mathrm{d}t} = -\tilde{x}_1 + (1-\tilde{x}_1^2)\tilde{x}_2$ and observation equation $\tilde{y} = x_1$. We denote this system as Σ_2. The input–output equation of Σ_2 is $\frac{\mathrm{d}^2\tilde{y}}{\mathrm{d}t^2} + \tilde{y} - (1-\tilde{y}^2)\frac{\mathrm{d}\tilde{y}}{\mathrm{d}t} = 0$. By dividing the left-side of this by F_1 following the procedures explained in Example C.1 in Supplementary material, we obtain the reminder R_1 as follows:

$$R_1 := (1-\theta_1)\frac{\mathrm{d}y}{\mathrm{d}t}y^2 - \frac{\theta_4}{3}y^3 + \frac{\theta_1^2 - \theta_1 - \theta_4}{\theta_1}\frac{\mathrm{d}y}{\mathrm{d}t} + \theta_4 y + (\theta_2\theta_4 + \theta_3). \tag{41}$$

Based on R_1 and Theorem 2, it is shown that Σ_1 generates a time-series generated by Σ_2 if, essentially, it holds that

$$(\theta_1, \theta_3, \theta_4)^\top = (1, 0, 0)^\top \tag{42}$$

given appropriately chosen initial values of (39) such that the consistent outputs $\hat{y}(t;\theta,u,x(0)), \dots, \frac{\mathrm{d}^2}{\mathrm{d}t^2}\hat{y}(t;\theta,u,x(0))$ evaluated at $t = 0$ are

equal to $\bar{y}(0), \ldots, \left.\frac{\mathrm{d}^N \bar{y}}{\mathrm{d}t^N}\right|_{t=0}$. Specifically, considering that $y = x_1$ in Σ_1, we solve the following systems of equations

$$\begin{pmatrix} \bar{y}(0) \\ \left.\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\right|_{t=0} \\ \left.\frac{\mathrm{d}^2\bar{y}}{\mathrm{d}t^2}\right|_{t=0} \end{pmatrix} = \begin{pmatrix} x_1(0) \\ \left.\frac{\mathrm{d}x_1}{\mathrm{d}t}\right|_{t=0} \\ \left.\frac{\mathrm{d}^2x_1}{\mathrm{d}t^2}\right|_{t=0} \end{pmatrix} \tag{43}$$

with respect to $x(0) = (x_1(0), x_2(0))^\top$ noting that the following holds:

$$\begin{aligned}
\left.\frac{\mathrm{d}x_1}{\mathrm{d}t}\right|_{t=0} &= \theta_1 \left(x_1(0) - \frac{1}{3}x_1(0)^3 + x_2(0) + \theta_2\right), \\
\left.\frac{\mathrm{d}x_2}{\mathrm{d}t}\right|_{t=0} &= -\frac{1}{\theta_1}\left(x_1(0) - \theta_3 + \theta_4 x_2(0)\right), \\
\left.\frac{\mathrm{d}^2x_1}{\mathrm{d}t^2}\right|_{t=0} &= \theta_1 \left(\left.\frac{\mathrm{d}x_1}{\mathrm{d}t}\right|_{t=0} - x_1(0)^2 \left.\frac{\mathrm{d}x_1}{\mathrm{d}t}\right|_{t=0} + \left.\frac{\mathrm{d}x_2}{\mathrm{d}t}\right|_{t=0}\right), \\
\left.\frac{\mathrm{d}^2x_2}{\mathrm{d}t^2}\right|_{t=0} &= -\frac{1}{\theta_1}\left(\left.\frac{\mathrm{d}x_1}{\mathrm{d}t}\right|_{t=0} + \theta_4 \left.\frac{\mathrm{d}x_2}{\mathrm{d}t}\right|_{t=0}\right).
\end{aligned}$$

Given these constraints on the parameters and initial values, Σ_1 is shown to be replaceable with Σ_2. To numerically confirm this, we conduct the following comparative experiments, in which the value of θ_2 is taken from $[0, 1]$.

Experiment 3-1. *Replaceability with respect to a single target time-series.* To demonstrate how the proposed method works numerically, we first consider the target system Σ_2 given an initial state $(\tilde{x}_1(0), \tilde{x}_2(0))^\top = (2, 0)^\top$, and thus, consider the replaceability with respect to a target time-series. In Fig. 3(a), Σ_1 is designed such that it satisfies the constraints on the parameters (42) and (43). In particular, the values of the parameters are as follows:

$$(\theta_1, \theta_2, \theta_3, \theta_4)^\top = (1, 0.5, 0, 0)^\top. \tag{44}$$

The details on the derivation of the consistent initial condition $x(0) = (2, 0.167)^\top$ is described in . As it follows the proposed design guidelines, the numerically-computed output of Σ_1 with Σ_2 is well-fitted, and the average of the normalized MSE is 0.0056. In contrast, Figs. 3 (b) and (c) show that the results with Σ_1 disobey some of these guidelines, in particular, without the constraints on the (b) parameters and (c) initial states. As shown in Figs. 3 (b) and (c), when neither (42) nor $x(0) = (2, 0.167)^\top$ is unsatisfied, the outputs of Σ_1 and Σ_2 greatly differ. The average values of the normalized MSE were 1.0395 and 0.0983. We also consider the case in which the target system Σ_2 lies in its stationary state; $t \in [100, 200]$. At this duration, states of Σ_2 forms an attractor, as shown in the red line in Fig. 3(d). Considering that Σ_2 is originally a two-dimensional system, the attractor is reconstructed in a two-dimensional delayed coordinate. The time delay was set so that the autocorrelation function of a discretized time-series sampled from the target time-series takes the first local minima [39] . Owing to the replaceability, the only condition left for the Σ_1 to generate the target time-series in stationary time could be (42). Σ_1 under the similar condition as that in Fig. 3(c) can successfully reconstruct similar attractors, as shown in Fig. 3(d). Thus, we confirmed that Σ_2 can be replaced by Σ_1 under the design guideline (42) with initial conditions $x(0) = (2, 0.167)^\top$ for the stationary task, and Eq. (42) for the non-stationary task. Note that both the stationary and non-stationary tasks are accomplished using our algebraic approach.

Experiment 3-2. *Replaceability with respect to initial values of a target system.* For simplicity, we consider the Σ_1 that has the same parameter values, (44). Here, we randomly choose $(\tilde{x}_1(0), \tilde{x}_2(0))^\top$ from $[-2, 2]^2$ 10 times and design the initial states of Σ_1 so that it they fit each target time-series. The results are summarized in Fig. 4(a) . The normalized MSE is 3.8173×10^{-6}. As observed, when the initial values of Σ_1 are appropriately chosen, Σ_1 approximates the output of Σ_2 well regardless of the $(\tilde{x}_1(0), \tilde{x}_2(0))^\top$. This indicates that the Σ_1 satisfying the proposed design guidelines, that takes consistent initial values and satisfies the parameter constraints, can be replaced with Σ_2 for various initial states.

Experiment 3-3. *Replaceability with respect to parameters of a target system.* We consider a modified target system, denoted by Σ_2'. Differential equations of Σ_2' are

$$\frac{\mathrm{d}\tilde{x}_1}{\mathrm{d}t} = \tilde{x}_2, \ \frac{\mathrm{d}\tilde{x}_2}{\mathrm{d}t} = -\tilde{x}_1 + \tilde{\theta}(1 - \tilde{x}_1^2)\tilde{x}_2 \tag{45}$$

where $\tilde{\theta} \in \mathbb{R}$ denotes its parameter and $\tilde{x}_1$ is observed. The input–output equation of Σ_2' is $\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + y - \tilde{\theta}(1 - y^2)\frac{\mathrm{d}y}{\mathrm{d}t}$. The remainder of this by F_1, the left-side of (40), with respect to the input–output order, is as follows:

$$(-\theta_1 + \tilde{\theta})\frac{\mathrm{d}y}{\mathrm{d}t}y^2 + \frac{{\theta_1}^2 - \theta_1\tilde{\theta} - \theta_4}{\theta_1}\frac{\mathrm{d}y}{\mathrm{d}t} - \frac{\theta_4}{3}y^3 + \theta_4 y + (\theta_2\theta_4 + \theta_3). \tag{46}$$

Here, we let $\tilde{\theta}$ take 1,3, and -3 and set $(\theta_1, \theta_2, \theta_3, \theta_4)^\top$ as $(\tilde{\theta}, 0.5, 0, 0)^\top$. The same initial values of the target system were applied as previous experiments and $x(0) = (x_1(0), x_2(0))$ is set to $(2, 0.167)^\top$. As observed in Fig. 4(b), if Σ_1 satisfies $(\theta_1, \theta_3, \theta_4)^\top = (\tilde{\theta}, 0, 0)^\top$, it can be replaced with and approximate Σ_2'. The average of the normalized MSE was 1.7765×10^{-6}.

5. Conclusion

In this paper, we proposed an algebraic design framework for physical computing systems that can offer specific design guidelines. The proposed framework, unlike practical design methods [12,13], is supported by rigorous theory, specifically, algebraic theory; unlike existing theories based on nonlinear filter approximation [14,16], it provides specific design guidelines; and unlike existing theories based on feedback linearizability [15,20], the applicability is not essentially limited — it is applicable regardless of whether it is feedback linearizable or not. Theorems 1 and 2 are presented as the main results followed by numerical experiments implementing our framework for solving several time-series generation tasks and replaceability-related tasks. It is noteworthy that the proposed framework is supported by computer algebraic techniques, and thus, the users can

Theorem 1 provides specific design guidelines on inputs given physical system generates a target time-series. To do this, the input–output map of a given physical computing system, which can be fortunately obtained using computer algebra software, is connected to target tasks. Thanks to the explicit description of the input–output maps based on algebra, specific design guidelines are successfully obtained. This is demonstrated through Experiments 1 and 2 in the context of time-series generation tasks. The derivation of specific design guidelines is the fundamental progress from existing approaches such as learning-based methods [12,13] and existing theories based on filter approximation theories [14,16]. Interestingly, our proposed approach shows that the physical computing system considered in Experiment 1 is differentially flat, meaning that it can be transformed into a linear systems with a certain feedback control. Differential flatness is beneficial property because, with appropriate observation equations, the system can be transformed into a linear system, for which control methods are well-developed and practically feasible. This fact suggests that the proposed algebraic approach has the potential to be practically applicable. In fact, problem settings of feedback controls are common for physical computing [14,15], and thus, combining flatness-based control and our approach is promising, which we leave as future work. Although we emphasized the connection to differential flatness, our approach is applicable regardless of whether the system is differentially flat, as demonstrated through Experiment 2 in Supplementary material.

Theorem 2 considers the replaceability of given physical computing systems to target systems. By relating the input–output maps of a given physical system and target systems, the replaceability is discriminated algebraically and specific design guidelines are provided if it is shown

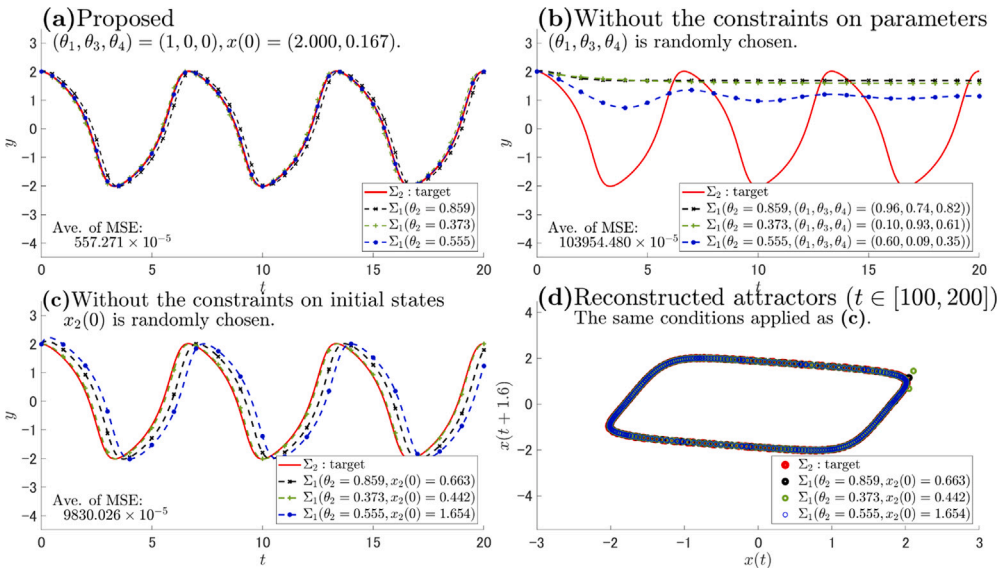

Fig. 3. Replaceability of FitzHugh–Nagumo system (Σ_1) with respect to the target system (Σ_2) given constraints on the **(a)** parameters and initial states, **(b)** initial states, and **(c)** parameters. **(d)** The reconstructed attractors under the same conditions as **(c)**.

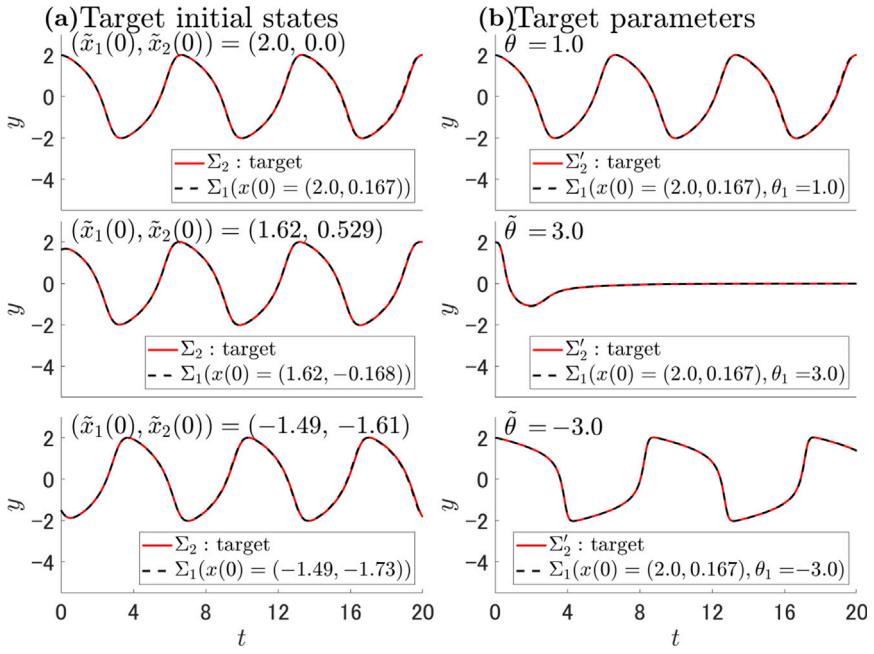

Fig. 4. Replaceability of Σ_1 with respect to $(\tilde{x}_1(0), \tilde{x}_2(0))^\top$ or $\tilde{\theta}$ of Σ_2. The outputs of Σ_1 are shown in dashed lines and the target time-series are shown in solid red lines.

to be replaceable. In Experiment 3, the replaceabilities concerning initial values as well as parameters of a target system are investigated. It is noteworthy that designing replaceability system is demanded but not well examined in the field of physical computing. This implies room for further contribution of the algebraic design framework for physical computing.

One of the limitations of our framework is related to the computational costs for the input–output equations; the Gröbner basis computation is often computationally expensive for large physical computing systems. Incorporating efficient methods for deriving input–output equations such as [40], which is recently proposed, into our framework may be a promising method to avoid such a scale limitation.

In this study, we assumed that physical computing system is modeled by polynomial differential equations and polynomial observation equations. However, some systems are not described in these types of equations. The existence of system noise and observation noise is also of concern. Thus, from a technical viewpoint, the expansion of the scope of our framework in terms of models is needed. In conventional approaches [14,15,17] based on approximation theory of filters [17], the Stone–Weierstrass theorem [41] is one of the keys. In the theorem, errors between a continuous function that corresponds to a considered filter and polynomial functions that approximate the filter are evaluated. Thereafter, it is shown that there exists a polynomial function that approximates the function representing the filter sufficiently under the appropriate assumptions. This implies that algebraic frameworks may be applicable to models described by continuous functions as well. This may broaden the scope of our frameworks. Therefore, if algebraic approaches are combined with these conventional approaches, designing

methods of physical computing system that are practically applicable may be constructed, which we leave for future work.

Furthermore, applying the proposed framework to more practical tasks remains a challenge to be addressed moving forward. In Experiment 3, we confirmed that algebraic design allows physical computing systems to perform non-stationary tasks. Although it should be noted that imposing initial conditions as demonstrated may be practically challenging, novel applications of physical computing systems may be considered based on our results. For instance, sequence-to-sequence tasks that appear in the field of natural language processing are in our scope in principle. It is important to apply algebraic design to these applications while considering the above technical considerations, which we leave for future work.

CRediT authorship contribution statement

Mizuka Komatsu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Takaharu Yaguchi:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Kohei Nakajima:** Writing – review & editing, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by JST CREST, Japan Grant JPMJCR1914, PRESTO, Japan Grants JPMJPR15E7 and JPMJPR16EC, ASPIRE, Japan Grant JPMJAP2329, ACT-X, Japan Grant JPMJAX22A7, Japan. M. K is also supported by JSPS, Japan KAKENHI Grand Number 22K21278 and 24K16963. K.N. is also supported by JSPS, Japan KAKENHI Grant Number JP18H05472 . This work is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Appendix A. Details of Experiments 1 and 3: estimations of consistent initial parameters

Theorem 1 and Theorem 2 require the consistent initial values defined in Definition 13 and the consistent initial output defined as (28). As explained soon after Definition 13, for arbitrary initial values of (12a), say $x(0)$, consistent initial values $x(0),\dots,\left.\frac{\mathrm{d}x}{\mathrm{d}t}\right|_{t=0}$ can be obtained. Specifically, $\left.\frac{\mathrm{d}^i x}{\mathrm{d}t^i}\right|_{t=0}$ $(i=1,\dots,N)$ can be obtained by using $x(0),\dots,\left.\frac{\mathrm{d}^{i-1}x}{\mathrm{d}t^{i-1}}\right|_{t=0},u(0),\dots,\left.\frac{\mathrm{d}^N u}{\mathrm{d}t^N}\right|_{t=0}$. Thus, according to Lemma 2, the output and its derivatives up to Nth order $\hat{y}(t;\theta,u,x(0)),\dots,\left.\frac{\mathrm{d}^N}{\mathrm{d}t^N}\hat{y}(t;\theta,u,x(0))\right|_{t=0}$ are uniquely obtained for arbitrary $x(0)$. Considering this, given $\bar{y}\in\mathcal{F}_y$ and $u\in\mathcal{F}_u$, to make initial values of (14a) and outputs of (14b) consistent, it suffices to set the initial values of (12a) as the solution of (28) with respect to $x(0)$. See a specific example in Experiment 1.

In Experiment 3, the initial values of (39) must be fixed consistently depending on initial values of Σ_2. Here, we consider the case that initial values of van der Pol equation, which is a part of Σ_2, are known. In this case, we first estimate the values of $\left.\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\right|_{t=0}$ and $\left.\frac{\mathrm{d}^2\bar{y}}{\mathrm{d}t^2}\right|_{t=0}$ via Chebyshev approximation similar as in Experiment in Section 4.2. The approximated values of $\left.\frac{\mathrm{d}\bar{y}}{\mathrm{d}t}\right|_{t=0}$ and $\left.\frac{\mathrm{d}^2\bar{y}}{\mathrm{d}t^2}\right|_{t=0}$ together with known $\bar{y}(0)$ are then substituted into (43). In this procedure, $\bar{y}$ is numerically computed by *the chebop function* of *chebfun* . To perform polynomial interpolation, we apply *the interp1 function* of *chebfun*. (43) is solved as a nonlinear least-squares problem by the Levenberg-Marquardt method, which estimates the initial values of Σ_1.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.physd.2024.134382.

References

[1] H. Jaeger, Towards a generalized theory comprising digital, neuromorphic and unconventional computing, Neuromorphic Comput. Eng. 1 (1) (2021) 012002.
[2] H. Yasuda, P.R. Buskohl, A. Gillman, T.D. Murphey, S. Stepney, R.A. Vaia, J.R. Raney, Mechanical computing, Nature 598 (2021) 39—48.
[3] C. Kaspar, B. Ravoo, W.G. van der Wiel, S.V. Wegner, W.H.P. Pernice, The rise of intelligent matter, Nature 594 (2021) 345–355.
[4] K. Nakajima, Physical reservoir computing—an introductory perspective, Japan. J. Appl. Phys. 59 (6) (2020) 060501.
[5] D. Kobayashi, Y. Kakehashi, K. Hirose, S. Onoda, T. Makino, T. Ohshima, S. Ikeda, M. Yamanouchi, H. Sato, E.C.I. Enobio, T. Endoh, H. Ohno, Influence of heavy ion irradiation on perpendicular-anisotropy cofeb-mgo magnetic tunnel junctions, IEEE Trans. Nucl. Sci. 61 (2013) 1710–1716.
[6] N. Akashi, Y. Kuniyoshi, S. Tsunegi, T. Taniguchi, M. Nishida, R. Sakurai, Y. Wakao, K. Kawashima, K. Nakajima, A coupled spintronics neuromorphic approach for high-performance reservoir computing, Adv. Intell. Syst. 4 (10) (2022) 2200123.
[7] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.
[8] H. Fang, A. Shrestha, Q. Qiu, Multivariate time series classification using spiking neural networks, in: 2020 International Joint Conference on Neural Networks, IJCNN, 2020, pp. 1–7.
[9] M. Ghazal, A. Kumar, N. Garg, S. Pecqueur, F. Alibart, Neuromorphic signal classification using organic electrochemical transistor array and spiking neural simulations, IEEE Sens. J. 24 (6) (2024) 9104–9114.
[10] A.G. Hart, Generalised synchronisations, embeddings, and approximations for continuous time reservoir computers, Physica D 458 (2024) 133956.
[11] Z. Konkoli, S. Nichele, M. Dale, S. Stepney, Reservoir computing with computational matter, in: S. Stepney, S. Rasmussen, M. Amos (Eds.), Computational Matter, Springer International Publishing, Cham, 2018, 269–293.
[12] L.G. Wright, T. Onodera, M.M. Stein, T. Wang, D.T. Schachter, Z. Hu, P.L. McMahon, Deep physical neural networks trained with backpropagation, Nature 601 (2022) 549—555.
[13] M. Nakajima, K. Inoue, K. Tanaka, Y. Kuniyoshi, T. Hashimoto, K. Nakajima, Physical deep learning with biologically inspired training method: gradient-free approach for physical hardware, Nat. Commun. 13 (1) (2022) 7847.
[14] H. Hauser, A.J. Ijspeert, R.M. Füchslin, R. Pfeifer, W. Maass, Towards a theoretical foundation for morphological computation with compliant bodies, Biol. Cybern. 105 (2011) 355–370.
[15] W. Maass, P. Joshi, E.D. Sontag, Computational aspects of feedback in neural circuits, PLoS Comput. Biol. 3 (1) (2007) e165.
[16] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.
[17] S. Boyd, L. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series, IEEE Trans. Circuits Syst. 32 (11) (1985) 1150–1161.
[18] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks, in: German National Research Center for Information Technology GMD Technical Report, 148, 2001.
[19] G. Manjunath, H. Jaeger, Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks, Neural Comput. 25 (3) (2013) 671–696.
[20] H. Hauser, A.J. Ijspeert, R.M. Füchslin, R. Pfeifer, W. Maass, The role of feedback in morphological computation with compliant bodies, Biol. Cybernet. 106 (10) (2012) 595–613.
[21] M. Fliess, J. Lévine, P. Martin, P. Rouchon, Sur les systèmes non linéaires différentiellement plats, C. R. Acad. Sci. Paris 315 (1992) 619—624.

[22] M. Fliess, J. Lévine, P. Martin, P. Rouchon, Flatness and defect of non-linear systems: introductory theory and examples, Internat. J. Control 61 (6) (1995) 1327–1361.
[23] M. Fliess, J. Lévine, P. Martin, P. Rouchon, A Lie-backlund approach to equivalence and flatness of nonlinear systems, IEEE Trans. Autom. Control. 44 (5) (1999) 922–937.
[24] G.G. Rigatos, Nonlinear control and filtering using differential Flatness Approaches applications to electromechanical systems, Springer Cham, 2015.
[25] D.A. Cox, J. Little, D. O'Shea, Ideals, varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, Springer, Cham, 2015.
[26] M. Komatsu, T. Yaguchi, K. Nakajima, Algebraic approach towards the exploitation of "softness": the input–output equation for morphological computation, Int. J. Robot. Res. 40 (1) (2021) 99–118.
[27] K. Nakajima, I. Fischer, Reservoir Computing: Theory, Physical Implementations, and Applications, Springer-Singapore, 2021.
[28] K. Forsman, Constructive Commutative Algebra in Nonlinear Control Theory in: Linköping Studies in Science and Technology. Dissertations, (261) Linköping University, 1991, 192 pages.
[29] V. Weispfenning, Comprehensive gröbner bases, J. Symbolic Comput. 14 (1) (1992) 1–29.
[30] D. Kapur, D. Wang, A new algorithm for computing comprehensive gröbner systems, in: Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC, 2010, pp. 29–36.
[31] D. Kapur, Comprehensive gröbner basis theory for a parametric polynomial ideal and the associated completion algorithm, J. Syst. Sci. Complex. 30 (2017) 196–233.
[32] L.H. Nguyen, K.-S. Hong, Synchronization of coupled chaotic FitzHugh–nagumo neurons via Lyapunov functions, Math. Comput. Simulation 82 (4) (2011) 590–603.
[33] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[34] O. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.
[35] L.N. Trefethen, Approximation Theory and Approximation Practice, Extended Edition, SIAM-Society for Industrial and Applied Mathematics, 2019.
[36] P. Brunovský, A classification of linear controllable systems, Kybernetika 6 (3) (1970) 173–188.
[37] G. Rigatos, P. Siano, N. Zervos, A new concept on flatness-based control of nonlinear dynamical systems, in: 2015 IEEE 13th International Conference on Industrial Informatics, INDIN, 2015, pp. 1146–1152.
[38] B. van der Pol, Relaxation-oscillations, Lond. Edinb. Dublin Philos. Mag. J. Sci. 7 (1926) 978–992.
[39] M. Mannattil, A. Pandey, M.K. Verma, S. Chakraborty, On the applicability of low-dimensional models for convective flow, Eur. Phys. J. B 90 (259) (2017) 9 pages.
[40] R. Dong, C. Goodbrake, H.A. Harrington, G. Pogudin, Differential elimination for dynamical models via projections with applications to structural identifiability, SIAM J. Appl. Algebra Geom. 7 (1) (2023) 194–235.
[41] M. Stone, The generalized weierstrass approximation theorem, Math. Mag. 21 (4) (1948) 167–184.