International Journal of Control, Automation, and Systems 23(1) (2025) 249-261
http://dx.doi.org/10.1007/s12555-023-0517-1
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

# A Data-driven Koopman Modeling Framework With Application to Soft Robots

**Lvpeng Han, Kerui Peng, Wangxing Chen, and Zhaobing Liu***

**Abstract:** This paper presents a data-driven Koopman modeling framework for globally linearizing highly nonlinear dynamical systems in lifted infinite-dimensional state space. In this framework, three data-driven models are proposed and identified to approximate the infinite-dimensional linear Koopman operator through a method called extended dynamic mode decomposition (EDMD). The implementation of EDMD requires a data set of snap pairs and a dictionary of scalar observables, which affects the accuracy of data-driven modeling. Five basis functions are compared and discussed to illustrate their suitable application scenarios. To verify the Koopman data-driven modeling framework, we apply it to the modeling of soft robotic systems, which is thought to be an extremely difficult task due to the large and continuous deformation of soft materials. Results demonstrate that Koopman linear, bilinear, and nonlinear models for both two-dimensional (2D) and three-dimensional (3D) soft robots are superior to the existing state-space modeling approach by achieving less normalized root mean square error (NRMSE). Among the three Koopman models, the constructed nonlinear model has higher performance than the bilinear model, followed by the linear one. Furthermore, the Monomial and Hermite basis functions are the optimal choices for constructing the Koopman linear, bilinear, and nonlinear models for the investigated 2D and 3D soft robots as they have the same structure when the degree of basis function is chosen less than four. Although the Fourier basis function is expected to outperform the Hermite and Gaussian basis functions in modeling systems with oscillatory motion, it is not a preferable selection in the high dimensional soft robotic modeling in our case due to its computational complexity and tendency to be non-convergent. The Sparse Fourier basis function cannot be regarded as a good choice for modeling soft robots, as it is only suitable for generating models with sparse data. It is worth noting that our findings can lay a solid foundation for the dynamics analysis and precise control of highly nonlinear dynamical systems, like soft robots in the future.

**Keywords:** Data-driven modeling, identification method, Koopman operator, nonlinear systems, soft robots.

## 1. INTRODUCTION

Linearization of nonlinear dynamical systems is one of the most well-known research topics in systems and control. In particular, in robotic systems, their dynamics are usually unknown, high dimensional, and highly nonlinear [1]. This makes it very difficult to obtain their models accurately, thus bringing great challenges to dynamics analysis and real-time control. In this regard, if exact linear equivalents or approximations of a nonlinear dynamical system are found, then standard techniques for linear systems can be used to analyze the complex behaviors of these systems.

Traditionally, for complex nonlinear systems, models coming from first principles are not available or do not fully resemble the original system due to unknown dynamics. Although systems can be represented in the state space form by the local linearization method based on Taylor series expansion, which achieves some success, they have limitations when it comes to efficient prediction, analysis, and control. Regarding this point, in recent years, scholars have shifted their focus to the Koopman operator theory, which had been proposed by Koopman in 1931 [2]. The idea behind this theory is that a finite-dimensional nonlinear system can be represented by an infinite-dimensional linear one without loss of accuracy by evolving functions of the states. For example, some soft robots may have infinite degrees of freedom (DoF), but the DoF of most conventional rigid robots is finite. A larger number of DoFs implies more complex and coupled nonlinearity. The Koopman operator theory is proven to be one of the optimal choices for modeling robots as

Manuscript received August 10, 2023; revised January 21, 2024, April 11, 2024, and August 19, 2024; accepted November 8, 2024. Recommended by Associate Editor Pilwon Hur under the direction of Senior Editor Dongjun Lee. The authors would like to thank the Start-up Funds from Wuhan University of Technology and the National Innovation and Entrepreneurship Training Program for College Students (S202410497076) for undertaking this work. Lvpeng Han and Kerui Peng are co-first authors.

Lvpeng Han, Kerui Peng, Wangxing Chen, and Zhaobing Liu are with the School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China (e-mails: {2896429913, 1135697976, 2259584379}@qq.com, zhaobingliu@whut.edu.cn).
* Corresponding author.

it can aid in resolving both the difficulty with nonlinearity and the requirement of data-driven modeling. Otto and Rowley [3] discussed the theoretical foundations of Koopman operator methods, as well as numerical methods developed over the past two decades to approximate the Koopman operator from data, for systems both with and without actuation. Zhang *et al*. [4] presented a robust tube-based MPC solution with Koopman operators. Fundamental properties, e.g., stabilizability, observability, of the Koopman model were derived under standard assumptions with which, the closed-loop robustness and nominal point-wise convergence were proven. Bruder *et al*. [5] investigated a method to construct an explicit dynamics model for soft robots based on Koopman operator theory and achieved relatively satisfactory modeling results. In the further work of Bruder *et al*. [6], the bilinear model of soft robots can be realized by the Koopman operator from input data. Kamenar *et al*. [7] utilized this approach to model the soft robot and predict its motion. In the study of Giorgos *et al*. [8], this method was utilized to model the tail-actuated robotic fish and achieved satisfying results. This approach can also be used to obtain a linear representation of underwater soft robots [9]. Besides, this theory has been investigated by many other scholars and applied in the modeling of other fields, such as modeling high-speed trains [10], wind farms [11], power grid applications [12], micro-aerial vehicles [13], and so on.

Given that the Koopman operator is infinite-dimensional, unless one can discover a finite-dimensional invariant subspace, suitable finite-dimensional approximations to the operator that can capture the system dynamics with high fidelity, are necessitated. For this reason, many data-driven approaches have been explored to achieve the finite-dimensional realizations of the Koopman operator. One of the popular data-driven methods to approximate the Koopman operator is called extended dynamic mode decomposition (EDMD) proposed in [14]. The finite-dimensional approximation generated by EDMD is the solution to a least-squares problem and converges to a Galerkin method with a large amount of data. The dimensionality and basis functions selected determine the modeling accuracy of the linear representation. That is to say, the choice of basis functions poses a critical challenge to models using Koopman operator theory [15]. To date, there is still no systematic way to address the selection problem of basis functions. In the existing literature, most studies rely on

1) Trial-and-error methods. For example, in the process of obtaining the Koopman model of high-speed trains, Chen *et al*. [10] selected the Gaussian function as the basis function by trial and error method. In [16,17], researchers also utilized this method to select polynomials as the basis function to obtain the Koopman model. This method may require time and effort and cannot guarantee generalization and efficiency in new cases.
2) Machine learning tools. The EDMD method can be improved by introducing the idea of machine learning. Specifically, the basis functions are trained and represented by the artificial neural network in the process of approximating the Koopman operator [18-20]. Shi and Meng [21] proposed an end-to-end deep learning framework that trained the embedding function together with the Koopman operator while maintaining the prediction quality and consistency of the cost function in the embedding space. It also modeled state-dependent nonlinearities in the control inputs, improving the prediction performance while preserving the linear dynamics of the modeled system and making linear control methods feasible. This is a combination of deep learning and Koopman that avoids the choice of basis functions, but this method is difficult and computationally intensive to find a unique inverse mapping function to recover the true control signal from the output of its auxiliary control network. Although the introduction of machine learning methods for soft robot systems can aid in the selection of the basis function, one unavoidable drawback of this approach is the need to make multiple complex adjustments (e.g., the number of layers, number of units per layer of neural networks, etc.), and large amounts of training data in the actual modeling process. This may pose significant challenges in robotic applications where data is small in principle.
3) System-specific methods. In terms of Koopman's modeling of rigid robots, the selection of basis functions is mostly based on the properties and composition of the dynamical equations of the system [22]. For instance, for micro-aerial vehicles, the basis function is mainly composed of parameters such as angular velocity and linear velocity [13]. Two joint angles and their sine and cosine values, which are related to the dynamical equations, are chosen as basis functions in the modeling process of a rigid manipulator [23]. For a tail-actuated rigid robotic fish, the selection of the basis function depends on its system-specific parameters, including coordinates, angular velocity, linear velocity, and orientation [8]. Although this method can construct the basis functions of the Koopman operator and compute its error bounds using higher-order derivatives of nonlinear dynamics that do not need to be known, this method is still not suitable for the calculation of error bounds for other basis functions. Besides, there is a type of spherical robot in which the selection of the basis function is analogous to the criterion stated above [24]. In general, this system-specific selection method has obvious downsides, that is, there is uncertainty in the construction of the basis function and its dimensional se-

lection, which will significantly affect the precision of the developed model. In addition, Shi and Karydis [23] also proposed an extension of the system-specific approach that needs system properties such as configuration space, and its workspace to formalize the analytical construction for dictionaries of lifting functions in the EDMD algorithm, which they call ACD-EDMD. Specifically, the basis function is formed by the first-order Hermite polynomial and its Kronecker product. However, when the basis function and its inner product are selected as the basis function form, this method may not guarantee that the Hermite matrix is the best choice. Furthermore, this method is only applicable to the modeling of 2D robots but not to 2.5D or 3D robots.

Although the existing work has achieved some beneficial results, to the best of the authors' knowledge, there is still a lack of a unified framework integrating the different data-driven models with suitable basis functions to realize the approximations of the Koopman operator. Given this, we present a data-driven Koopman modeling framework for globally linearizing highly nonlinear dynamical systems in lifted infinite-dimensional state space. In this framework, three data-driven models (linear, bilinear, and nonlinear) are developed and identified to approximate the infinite-dimensional linear Koopman operator through the method of EDMD. To obtain the above data-driven models with high performance, five basis functions are evaluated and discussed to demonstrate their suitable scenarios, which resolves the inadequacy of the conventional trial-and-error methods while giving guided suggestions on their selection. To verify the proposed methodology, it has been applied to the modeling and prediction of dynamics for both 2D and 3D soft robots.

The remainder of this paper is organized as follows: Section 2 introduces the data-driven Koopman modeling framework, including the Koopman operator, data-driven realization approaches, and the selection of basis functions. Section 3 presents the implementation of system identification, which mainly involves the experimental setup and model precision assessment. Then, the results are described, followed by a detailed discussion of the quantitative and qualitative analysis of the comparative methods. Finally, Section 4 summarizes the paper.

## 2. DATA-DRIVEN KOOPMAN MODELING FRAMEWORK

The Koopman operator exploits the principle that any finite-dimensional nonlinear system possesses an equivalent infinite-dimensional linear representation in the real-valued function space of system states and inputs. In the space of real-valued functions, this linear representation describes the flow of functions along the trajectory of the system. Although an infinite-dimensional Koopman operator cannot be represented numerically, it is possible to represent its projection onto a finite-dimensional subspace as a matrix. Moreover, the relationship between finite-dimensional and infinite-dimensional representations of a system is bijective and well-defined [5,6,14]. Therefore, this allows us to approximate the Koopman operator by linear regression on the observed data and then obtain the equivalent nonlinear system representation. This section presents a framework for approximating the Koopman operator from data and utilizing it to construct linear, bilinear, or nonlinear system models.

### 2.1. Theoretical background for Koopman operator

Consider a discrete-time nonlinear dynamic system with external inputs

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

in which $x_k \in X \subset \mathbb{R}^n$ and $u_k \in U \subset \mathbb{R}^m$ represent the state and input of the system, $f$ is the nonlinear dynamics of the system which evolves the system state in time, $k \in \mathbb{Z}$.

The Koopman operator advances the observables with the flow of the dynamics [5]

$$\mathcal{K}_d \psi = \psi \circ f, \tag{2}$$

where $\mathcal{K}_d$ denotes the discrete-time Koopman operator, c is the composition of $\psi$ with $f$, $\psi(x,u)$ is referred to as observables in the lifted space. Equation (2) can be further expressed as [25]

$$(\mathcal{K}_d \psi)(x_k, u_k) = \psi \circ f = \psi(x_{k+1}, u_{k+1}). \tag{3}$$

Equation (3) indicates that the application of the infinite-dimensional Koopman operator transitions a system from a finite dimensional nonlinear state space to an infinite-dimensional linear function space [25].

### 2.2. Data-driven approximation with EDMD algorithm

Since the Koopman operator is an infinite-dimensional object, we desire to obtain the projection of the Koopman operator on a finite-dimensional subspace. In this respect, the EDMD algorithm is utilized to identify a finite-dimensional matrix approximation of the Koopman operator to resolve this issue [14].

#### 2.2.1 EDMD algorithm

To derive an approximation of the Koopman operator $\bar{\mathcal{K}}_d \in \mathbb{R}^{N \times N}$, we use data to solve the least squares minimization problem which can be expressed by the following form

$$\bar{\mathcal{K}}_d^* = \arg\min_{\bar{\mathcal{K}}_d} \sum_{k=1}^{K} \frac{1}{2} \left\| \psi(x_{k+1}, u_{k+1}) - \bar{\mathcal{K}}_d \psi(x_k, u_k) \right\|^2, \tag{4}$$

where $\psi(x,u) = [\psi_1(x,u),\ \psi_2(x,u),\ ...,\ \psi_N(x,u)]^{\mathrm{T}}$ is a set of basis (observable) functions and $K$ represents the number of data measurements. Furthermore, the above expression can be rewritten by a closed-form solution

$$\bar{\mathcal{K}}_d^* = \psi_b \psi_a^{\dagger}, \tag{5}$$

in which

$$\begin{aligned} \psi_a &= \frac{1}{K}\sum_{k=1}^{K} \psi(x_k,u_k)\psi(x_k,u_k)^{\mathrm{T}}, \\ \psi_b &= \frac{1}{K}\sum_{k=1}^{K} \psi(x_{k+1},u_{k+1})\psi(x_k,u_k)^{\mathrm{T}}, \end{aligned} \tag{6}$$

and † denotes the Moore-Penrose pseudoinverse.

### 2.2.2 Koopman model realizations

• **Koopman linear model (KLM)**

Consider the Koopman representation as follows:

$$\psi(x_{k+1},u_{k+1}) = \bar{\mathcal{K}}_d \psi(x_k,u_k). \tag{7}$$

For simplicity, let $\psi(x,u) = [\psi_x^{\mathrm{T}}(x),\ \psi_u^{\mathrm{T}}(u)]^{\mathrm{T}}$, we see that $\psi_x(x)$ is independent with $u$ and $\psi_u(u)$ is independent with $x$. $\bar{\mathcal{K}}_d$ is reconstructed as the following form similar to that in [8]

$$\bar{\mathcal{K}}_d = \begin{bmatrix} A_{(N-m)\times(N-m)} & B_{(N-m)\times m} \\ O_{m\times(N-m)} & I_{m\times m} \end{bmatrix}. \tag{8}$$

Then, (7) can be rewritten as

$$\begin{bmatrix} \psi_x(x_{k+1}) \\ \psi_u(u_{k+1}) \end{bmatrix} = \bar{\mathcal{K}}_d \begin{bmatrix} \psi_x(x_k) \\ \psi_u(u_k) \end{bmatrix} \tag{9}$$

where $\psi_x(x) = [\psi_1(x),\ \psi_2(x),\ \ldots,\ \psi_{N-m}(x)]^{\mathrm{T}} \in \mathbb{R}^{N-m}$ and $\psi_u(u) = [\psi_{N-m+1}(u),\ \psi_{N-m+2}(u),\ \ldots,\ \psi_N(u)]^{\mathrm{T}} \in \mathbb{R}^m$ are the functions that depend on the system states and inputs, respectively. We use $\psi_u(u_k) = u_k$ to ensure that control appears linearly in the model such that $\psi_x(x_{k+1}) = A\psi_x(x_k) + Bu_k$.

When we choose a suitable set of basis functions $\psi = \{\psi_i\}_{i=1}^{N}$, (9) can be decomposable into a linear system as follows:

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k, \\ \hat{x}_k &= Cz_k, \end{aligned} \tag{10}$$

where $z_k = \psi_x(x_k)$ and $u_k = \psi_u(u_k)$. The output matrix $C = \begin{bmatrix} I_{n\times n} & O_{n\times(N-n)} \end{bmatrix}$.

To obtain the system matrices $A$ and $B$, substituting (10) into (4), it gets

$$\arg\min_{A,B} \sum_{k=1}^{K} \frac{1}{2} \left\| \psi_x(x_{k+1}) - A\psi_x(x_k) - Bu_k \right\|^2. \tag{11}$$

• **Koopman bilinear model (KBM)**

Similarly, let $\psi(x,u) = [\psi_x^{\mathrm{T}}(x),\ \psi_{x,u}^{\mathrm{T}}(x,u),\ \psi_u^{\mathrm{T}}(u)]^{\mathrm{T}}$, $\bar{\mathcal{K}}_d$ is reconstructed as the form

$$\bar{\mathcal{K}}_d = \begin{bmatrix} A_{N\times N} & \left\{H_{i(N\times N)}\right\}_{i=1}^{m} & B_{N\times m} \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{12}$$

Then, (7) can be rewritten as

$$\begin{bmatrix} \psi_x(x_{k+1}) \\ \psi_u(u_{k+1}) \end{bmatrix} = \bar{\mathcal{K}}_d \begin{bmatrix} \psi_x(x_k) \\ \psi_{x,u}(x_k,u_k) \\ \psi_u(u_k) \end{bmatrix}, \tag{13}$$

where $\psi_x(x) = [\psi_1(x),\ \psi_2(x),\ ...,\ \psi_N(x)]^{\mathrm{T}} \in \mathbb{R}^N$, $\psi_u(u) = [\psi_1(u),\ \psi_2(u),\ ...,\ \psi_m(u)]^{\mathrm{T}} \in \mathbb{R}^m$, and $\psi_{x,u}(x,u) = [\psi_x^{\mathrm{T}}(x)\psi_1(u),\ \psi_x^{\mathrm{T}}(x)\psi_2(u),\ ...,\ \psi_x^{\mathrm{T}}(x)\psi_m(u)]^{\mathrm{T}} \in \mathbb{R}^{\mathrm{Nm}}$.

When we choose a suitable set of basis functions $\psi = \{\psi_i\}_{i=1}^{N(m+1)+m}$, (13) can be decomposable into a bilinear system which can be expressed by the following

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k + \sum_{i=1}^{m} H_i z_k u_{ki}, \\ \hat{x}_k &= Cz_k, \end{aligned} \tag{14}$$

in which $z_k = \psi_x(x_k)$, $z_k u_k = \psi_{x,u}(x_k,u_k)$, and $u_k = \psi_u(u_k)$. The output matrix $C = \begin{bmatrix} I_{n\times n} & O_{n\times(N-n)} \end{bmatrix}$.

To get the system matrices $A$, $B$, and $H$, (4) can be rewritten as

$$\begin{aligned} \arg\min_{A,B,\{H_i\}_{i=1}^{m}} \sum_{k=1}^{K} \frac{1}{2} \Bigg\| \psi_x(x_{k+1}) - A\psi_x(x_k) - Bu_k \\ - \sum_{i=1}^{m} H_i \psi_x(x_k) u_{ki} \Bigg\|^2. \end{aligned} \tag{15}$$

• **Koopman nonlinear model (KNM)**

Let $\psi(x,u) = [\psi_x^{\mathrm{T}}(x),\ \psi_{x,u}^{\mathrm{T}}(x,u)]^{\mathrm{T}}$, $\bar{\mathcal{K}}_d$ is reconstructed as the form

$$\bar{\mathcal{K}}_d = \begin{bmatrix} A_{n\times n} & B_{n\times(N-n)} \\ * & * \end{bmatrix}. \tag{16}$$

Then, (7) can be rewritten as

$$\begin{bmatrix} \psi_x(x_{k+1}) \\ \psi_{x,u}(x_{k+1},u_{k+1}) \end{bmatrix} = \bar{\mathcal{K}}_d \begin{bmatrix} \psi_x(x_k) \\ \psi_{x,u}(x_k,u_k) \end{bmatrix}, \tag{17}$$

where $\psi_x(x) = [\psi_1(x),\ \psi_2(x),\ ...,\ \psi_n(x)]^{\mathrm{T}} \in \mathbb{R}^n$ and $\psi_{x,u}(x,u) = [\psi_{N-n+1}(x,u),\ \psi_{N-n+2}(x,u),\ ...,\ \psi_N(x,u)]^{\mathrm{T}} \in \mathbb{R}^{N-n}$. It is noted that a discrete nonlinear dynamical system with inputs can be formatted as

$$x_{k+1} = F(x_k,u_k), \tag{18}$$

where $F{:}X\times U \to \mathbb{R}^n$ is a nonl inear discrete map.

When we choose a suitable set of basis functions $\psi = \{\psi_i\}_{i=1}^{N}$, the nonlinear discrete map $F(x,u)$ in (18) can be further constructed by the Koopman operator $\bar{\mathcal{K}}_d$ as

$$F(x,u) = \begin{bmatrix} I_{n\times n} & O_{n\times(N-n)} \end{bmatrix} \bar{\mathcal{K}}_d \psi(x,u)$$

$$= A\psi_x(x) + B\psi_{x,\mu}(x,u), \tag{19}$$

in which, the system matrices $A$ and $B$ in $\bar{\mathcal{K}}_d$ can be solved by

$$\arg\min_{AB} \sum_{k=1}^{K} \frac{1}{2} \left\| \psi_x(x_{k+1}) - A\psi_x(x_k) - B\psi_{x,n}(x_k,u_k) \right\|^2. \tag{20}$$

### 2.2.3 Choice of basis functions

The choice of the basis function plays an essential role in approximating the Koopman operator, thereby affecting the quality of the resulting models and their prediction precision. Note that for different Koopman model realizations, the forms of basis functions are slightly different. For comparison purpose, we use the same notation $\psi$ to represent the basis functions.

As for the linear model realization, we define the basis functions as

$$\psi_i(\hat{x},u) = \begin{cases} h_i(\hat{x}), & \forall i \in \{1, \ldots, N-m\}, \\ u_i, & \forall i \in \{N-m+1, \ldots, N\}, \end{cases} \tag{21}$$

where $\psi_x = \mathbf{h}(\hat{x}) = [h_1(\hat{x}), ..., h_{N-m}(\hat{x})]^\intercal$ and $\psi_n = \mathbf{u} = [u_1, ..., u_m]^\intercal$.

As for the bilinear model realization, we define the basis functions as

$$\psi_i(\hat{x},u) = \begin{cases} h_i(\hat{x}), & \forall i \in \{1, \ldots, N\}, \\ h_{i-N}(\hat{x}) \cdot u_1, & \forall i \in \{N+1, \ldots, 2N\}, \\ \vdots & \vdots \\ h_{i-Nm}(\hat{x}) \cdot u_m, & \forall i \in \{Nm+1, \ldots, N(m+1)\}, \\ u_{i-N(m+1)}, & \forall i \in \{N(m+1)+1, \ldots, N(m+1)+m\}, \end{cases} \tag{22}$$

where $\psi_x = \mathbf{h}(\hat{x}) = [h_1(\hat{x}), ..., h_N(\hat{x})]^\intercal$, $\psi_n = \mathbf{u} = [u_1, ..., u_m]^\intercal$, and $\psi_{x,u} = \left[u_1\mathbf{h}^\mathrm{T}, ..., u_m\mathbf{h}^\mathrm{T}\right]^\mathrm{T}$.

In terms of the nonlinear model realization, we define the basis functions as

$$\psi_i(\hat{x},u) = \begin{cases} u_i, & \forall i \in \{1, \ldots, n\}, \\ h_i(\hat{x},u), & \forall i \in \{n+1, \ldots, N\}, \end{cases} \tag{23}$$

where $\psi_u = \mathbf{u} = [u_1, ..., u_n]^\intercal$, and $\psi_{x,u} = \mathbf{h}(\hat{x},u) = [h_{n+1}(\hat{x},u), ..., h_N(\hat{x},u)]^\intercal$.

As there is still no general criterion on how to choose an appropriate basis function, in this paper, five basis functions are evaluated and discussed to show their suitable scenario given the modeling process of soft robots. Specifically, they are the Monomial basis function, Hermite basis function, Gaussian basis function (also known as radial basis function), Fourier basis function, and Sparse Fourier basis function. It is worth noting that some of them have shown promising applications in the Koopman modeling of other nonlinear systems such as the high-speed train and wind farm. The selected basis functions and their dimensional equations are presented in Table 1.

Table 1. The expression and dimensional equations of the basis functions.

| Type | Expression | Dimension |
|---|---|---|
| Monomial | $x_1^{s_1}, x_2^{s_2}, \ldots, x_n^{s_n}$ $(s_1, s_2, \ldots, s_n \in \mathbb{N}^n)$ | $N = \frac{(m+1)\cdots(m+\rho)}{\rho!} - 1$ |
| Hermite | $H_1(x_1), H_2(x_2), \ldots$ | $N = \frac{(m+\rho)!}{m!\rho!} - 1$ |
| Gaussian | $\exp\left(-\frac{\|x-c_i\|^2}{\sigma_i^2}\right)$ $(c_i \in [-1, 1])$ | $N = \rho$ |
| Fourier | $f(x) = d_0 + \sum_{i=1}^{\infty} \left(d_i \cos\frac{i\pi x}{L} + e_i \sin\frac{i\pi x}{L}\right)$ | $N = (1+2\rho)^m - 1$ |
| Sparse Fourier | $x(n) = \sum_{i=1}^{L-1} d_i e^{-j\frac{2\pi}{L}nki}$ $(n \in [0, L-1])$ | $N = 4m + \frac{(2m)!}{\rho!(2m-\rho)!}$ |

Note: For Fourier and Sparse Fourier functions, $d_i$ and $e_i$ denote the Fourier coefficient. $L$ represents the length of the discrete-time signal. In the dimension equation of basis functions, $\rho$ represents the maximum degree of the basis function and $m$ denotes the states of the system. For instance, for linear and bilinear models, the value of $m$ is referred to as the number of $x_k, ..., x_{k-d}, u_{k-d}$, where $d$ represents the number of delays. In terms of the nonlinear model, $m$ is referred to as the number of $x_k$, ..., $x_{k-d}$, $u_{k-d}$.

Here, we give a detailed example to facilitate the understanding of how to build lifted space with five basis functions. In the example, the maximum degree $\rho = 2$, the time delay $d = 1$, and $m = 3$ with $x_2, x_1, u_1$.

- **Monomial basis function:** The monomials are the simple and good choice of polynomial vector fields [26]. The lifted vector with monomial basis functions is presented as follows:

$$\left[x_2,\ x_1,\ u_1,\ x_2^2,\ x_1^2,\ u_1^2,\ x_2x_1,\ x_2u_1,\ x_1u_1\right]_{9\times1}^{\mathrm{T}}. \tag{24}$$

- **Hermite basis function:** The Hermite polynomials can be categorized into probability and physics [27]. We consider the commonly-used physics Hermite polynomials as the basis function. The lifted vector with Hermite basis functions can be expressed as follows:

$$\left[x_2,\ x_1,\ u_1,\ x_2^2,\ x_1^2,\ u_1^2,\ x_2x_1,\ x_2u_1,\ x_1u_1\right]_{9\times1}^{\mathrm{T}}. \tag{25}$$

- **Gaussian basis function:** Gaussian basis function is a scalar function along the radial symmetry [28], which is suitable for radial oscillatory dynamics, such as a Duffing oscillator model. The lifted vector with Gaussian basis functions can be defined as follows:

$$\left[\exp\left(-\frac{\|\mathbf{x}-c_1\|^2}{\sigma_1^2}\right),\exp\left(-\frac{\|\mathbf{x}-c_2\|^2}{\sigma_2^2}\right)\right]^{\mathrm{T}}_{2\times 1}, \tag{26}$$

where $\mathbf{x}=[x_2, x_1, u_1]^{\mathrm{T}}$ and $\sigma_1=\sigma_2=1$.

• **Fourier basis function:** Compared with other basis functions, the Fourier basis function has the largest dimension and is computationally complex [29]. The lifted vector with Fourier basis functions can be defined as follows:

$$\begin{aligned}&[\sin 2\pi x_2,\ \sin 2\pi x_1,\ \sin 2\pi u_1,\ \cos 2\pi x_2,\ \cos 2\pi x_1,\\ &\cos 2\pi u_1,\ \sin 4\pi x_2,\ \sin 4\pi x_1,\ \sin 4\pi u_1,\ \cos 4\pi x_2,\\ &\cos 4\pi x_1,\ \cos 4\pi u_1,\ \sin 2\pi x_2 \sin 2\pi x_1,\\ &\sin 2\pi x_2 \cos 2\pi u_1,\ \ldots,\ \sin 4\pi x_1 \cos 4\pi u_1,\\ &\sin 2\pi x_2 \sin 2\pi x_1 \cos 2\pi u_1,\ \cos 2\pi x_2 \cos 2\pi x_1 \cos 2\pi u_1,\\ &\sin 2\pi x_2 \cos 2\pi x_2 \cos 2\pi x_1,\ \sin 2\pi x_2 \cos 2\pi x_2 \cos 2\pi u_1,\\ &\sin 2\pi x_2 \cos 2\pi x_1 \cos 2\pi u_1,\ \cos 2\pi x_2 \cos 2\pi x_1 \cos 2\pi u_1,\\ &\sin 2\pi x_1 \cos 2\pi x_2 \cos 2\pi x_1,\ \sin 2\pi x_1 \cos 2\pi x_2 \cos 2\pi x_1,\\ &\sin 2\pi x_1 \cos 2\pi x_2 \cos 2\pi x_1,\ \sin 2\pi x_1 \cos 2\pi x_2 \cos 2\pi x_1,\\ &\sin 2\pi x_1 \cos 2\pi x_1 \cos 2\pi x_1,\ \ldots]^{\mathrm{T}}_{128\times 1}.\end{aligned} \tag{27}$$

• **Sparse Fourier basis function:** The Sparse Fourier basis function is applied to the case with the sparsity of data [30]. This basis function has fast and simple calculations in the modeling process compared to the Fourier basis function. The lifted vector with Sparse Fourier basis functions can be defined as follows:

$$\begin{aligned}&[\sin 2\pi x_2,\ \sin 2\pi x_1,\ \sin 2\pi u_1,\ \cos 2\pi x_2,\ \cos 2\pi x_1,\\ &\cos 2\pi u_1,\ \sin 4\pi x_2,\ \sin 4\pi x_1,\ \sin 4\pi u_1,\ \cos 4\pi x_2,\\ &\cos 4\pi x_1, \cos 4\pi u_1, \sin 2\pi x_2 \sin 2\pi x_1, \sin 2\pi x_2 \sin 2\pi u_1,\\ &\sin 2\pi x_1 \sin 2\pi u_1, \cos 2\pi x_2 \cos 2\pi x_1, \cos 2\pi x_2 \cos 2\pi u_1,\\ &\cos 2\pi x_1 \cos 2\pi u_1, \sin 2\pi x_2 \cos 2\pi x_2, \sin 2\pi x_2 \cos 2\pi x_1,\\ &\sin 2\pi x_2 \cos 2\pi u_1, \sin 2\pi x_1 \cos 2\pi x_2, \sin 2\pi x_1 \cos 2\pi x_1,\\ &\sin 2\pi u_1 \cos 2\pi u_1, \sin 2\pi u_1 \cos 2\pi x_2, \sin 2\pi u_1 \cos 2\pi x_1,\\ &\sin 2\pi x_1 \cos 2\pi u_1]^{\mathrm{T}}_{27\times 1}.\end{aligned} \tag{28}$$

### 2.2.4 Data preparations and processing

To approximate the Koopman operator from a set of experimental data, we take $K$ discrete measurements to construct snapshot pairs $\{(a_k, b_k)\}$ and $u_k$, $k=1, ..., K$ which can be expressed by the following:

$$\begin{cases} a_k := x_k, \\ b_k := x_{k+1}. \end{cases} \tag{29}$$

Then, we can lift all of the snapshot pairs and compile them into the following $K\times N$ matrices

$$\psi_a := \begin{bmatrix} \psi(a_1,u_1)^{\mathrm{T}} \\ \psi(a_2,u_2)^{\mathrm{T}} \\ \vdots \\ \psi(a_K,u_K)^{\mathrm{T}} \end{bmatrix}, \quad \psi_b := \begin{bmatrix} \psi(b_1,u_1)^{\mathrm{T}} \\ \psi(b_2,u_2)^{\mathrm{T}} \\ \vdots \\ \psi(b_K,u_K)^{\mathrm{T}} \end{bmatrix}. \tag{30}$$

In this way, $\bar{\mathcal{K}}_d$ can be calculated by yielding the least-squares best fit for all of the observed data, which is given by

$$\bar{\mathcal{K}}_d := \psi_b \psi_a^{\dagger}. \tag{31}$$

In practice, the delays are incorporated into the set of snapshot pairs (29) to acquire a more accurate model, which can be expressed as

$$\begin{cases} \mathbf{a}_k = \left[x_k,\ ...,\ x_{k-d},\ u_{k-1},\ ...,\ u_{k-d}\right]^{\mathrm{T}}, \\ \mathbf{b}_k = \left[x_{k+1},\ ...,\ x_{k-d+1},\ u_{k-1},\ ...,\ u_{k-d}\right]^{\mathrm{T}}, \end{cases} \tag{32}$$

where $d$ represents the number of delays.

### 2.2.5 Implementation of the data-driven Koopman modeling framework

The data-driven Koopman modeling framework can be implemented through Algorithm 1.

**Remark 1:** In this paper, our contributions have two fold. Firstly, we've developed a unified framework that combines various data-driven models—linear, bilinear, and nonlinear—with appropriate basis functions to approximate the Koopman operator. Prior research has largely focused on linear model realizations using monomial basis functions, which limits modeling accuracy and, consequently, control performance. Secondly, we've refined the trial-and-error approach by standardizing the definition of basis functions. This allows us to integrate

**Algorithm 1:** Data-driven Koopman modeling process.

**Input:** Snapshot pairs $\{(a_k,b_k)\}$ and $u_k$, $k=1,\ldots,K$ and number of delays $d$

1) Choose a candidate of basis functions
2) **for** $\rho=1,2,\ldots$ **do**
3) Lift data via (30)
4) Approximate the Koopman operator $\bar{\mathcal{K}}_d$ via (31)
5) Obtain model matrices: $A, B$ for linear model via (8); $A$, $B$, $\{H_i\}_{i=1}^m$ for bilinear model via (12); $A$, $B$ for nonlinear model via (16)
6) Calculate NRMSE via (34)
7) **end for**
8) Derive the optimal basis function and its corresponding dimension $\rho$ based on the NRMSE results

**Output:** Linear, nonlinear and bilinear models with $A$, $B$, $C := \begin{bmatrix} I_{n\times n} & O_{n\times(N-n)} \end{bmatrix}$

and compare different basis functions within our algorithm, enabling us to identify the optimal one for enhanced modeling accuracy. This refinement gives our method an edge over the conventional trial-and-error technique.

## 3. RESULTS AND DISCUSSION

To test the method described in Section 2, we've applied it to soft robots that can deform continuously. We then compare these models to those created using state space system identification techniques. Below, we provide a detailed account of the robotic hardware used, the experimental setup, the process of data acquisition and processing for system identification, and how we evaluated and compared the performance of different models.

### 3.1. Experiments

#### 3.1.1 Experimental setup

In this paper, we use 2D and 3D soft robots to test and verify the performance of the Koopman data-driven modeling approach. The motion of soft robots is driven through the pressure regulation of the pneumatic supply from an air pump. The design of the pneumatic control module can refer to the soft robotics toolkit presented on the website (https://softroboticstoolkit.com/book/control-board). This module contains solenoid valves (SMC VQ110U-5M) that control the flow of the pneumatic supply to regulate the pressure using a pulse-width modulated (PWM) signal from an Arduino board (MEGA 2560). It provides a fairly simple and inexpensive method for controlling the pressure supply to the soft robots. The Arduino board interfaces with the pressure sensors (XGZP6847), the flex sensor (FS-L-0095-103-ST), and the positioning infrared radiation (IR) camera system (SEN0158) to measure the pneumatic pressure, bending angle, and X-Y plane coordinates of 2D and 3D robots. The sensory data were collected and recorded in a real-time manner in MATLAB/SIMULINK environment. The details of the experimental setup are illustrated in Fig. 1.

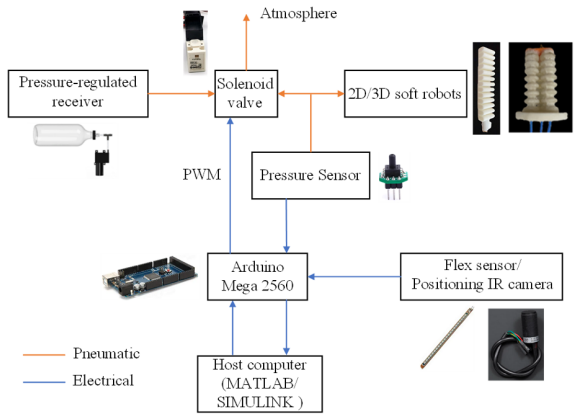



Fig. 1. Experimental setup.

#### 3.1.2 Data acquisition and processing

To construct the models, a set of experiments were carried out to obtain the input-output data pairs. For the 2D soft robot, 11460 input-output data pairs (10000 pairs are used as a training (modeling) set and the rest as a test (prediction) set) were used to identify the Koopman data-driven models, while for the 3D soft robot, 13955 pairs (12000 pairs are used as a training (modeling) set and the rest as a test (prediction) set). Note that, a method, similar to the one in [5], is used to generate the randomized inputs over the entire operating range, which is expressed by the following:

$$u_{k,j} = \frac{(\Upsilon_{i,j+1} - \Upsilon_{i,j})}{T}\left(kT_s + \frac{(i-1)T}{3}\right) + \Upsilon_{i,j}, \quad (33)$$

where $T_s = 0.1$ s is the sampling period, $T = 2, 3, 4, 5$ s is the transition time, and $j = \text{floor}\left(\frac{kT_s}{T}\right)$ represents the current index at time $t$. In addition, matrices $\Upsilon_{2D} \in [51.2, 153.6]^{1\times 1000}$ and $\Upsilon_{3D} \in [51.2, 102.4]^{3\times 1000}$ refer to the randomly distributed range of duty cycles for the 2D and 3D soft robots, respectively. It is noted that the values are determined according to the reasonable working pressures that soft robots can bear. The model of the 2D soft robot can be represented as a single-input and single-output model since it has only one air chamber. Its output is the bending angle measured by the flex sensor. While, for the 3D soft robot, a range of input pressures can be applied to the three different chambers to change the output position of the IR point on the $X$-$Y$ plane, in which the coordinates were captured by the positioning IR camera system.

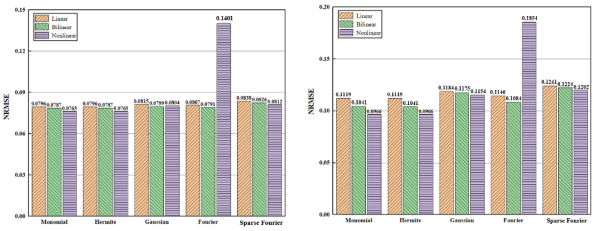
Here, we apply a normalization method that enables the input and output data to be normalized between $-1$ and 1. Then, the normalized data can be incorporated to obtain the Koopman data-driven models. Note that, data normalization enables an easier and more accurate capture of the correlations between the independent and dependent variables in different types of models, which can enhance the prediction precision of the data-driven models.

#### 3.1.3 Modeling precision assessment

Models, described in Section 2, are generated by using the collected training data sets. The precision (goodness of fit) of various models was evaluated by comparing the modeling and prediction results to each of the training and test data sets, which can be quantized using the normalized root-mean-square error (NRMSE)

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(s_{real} - s_{des})^2}}{\max(s_{real}) - \min(s_{real})}, \quad (34)$$

in which $m$ indicates the total number of data points; $s_{real}$ and $s_{ds}$ represent the measured and desired output values, respectively.




(a) (b)

Fig. 2. The comparison of modeling and performance of three Koopman models with five basis functions (2D soft robot): (a) Modeling and (b) prediction.

## 3.2. Case studies

In this section, we demonstrate and compare the modeling and prediction results of the established Koopman models utilizing five basis functions for the 2D and 3D soft robots in comparison to the state space model. Here, modeling results refer to the fitting accuracy when building the Koopman models using the training data. Prediction results refer to the generalization ability of the established Koopman models using the test data.

### 3.2.1 Case study 1: Modeling and prediction of 2D soft robot

• **Modeling**

For the 2D soft robot, we choose the maximum degree $\rho = 2$, the time delay $d = 1$, $m = 3$ for both linear and bilinear models, and $m = 4$ for the nonlinear model. The dimensions of the five basis functions used for constructing three Koopman models are summarized in Table 2, which is determined by the dimension equations as described in Table 1.

The comparison of the modeling results of three Koopman models developed by five basis functions and the state space model for the 2D soft robot is summarized in Table 3 and Fig. 2(a).

• **Prediction**

The comparison of the prediction results for the 2D soft robot obtained by three Koopman models with five basis functions is shown in Fig. 2(b).

To further demonstrate the prediction performance of the generated Koopman models, the evolution of the detailed prediction results is provided in Fig. 3, in which the basis function is chosen as the Monomial type.

The comparison of the modeling results of constructed three Koopman models and SSM for the 2D soft robot is quantitatively summarized in Table 4.

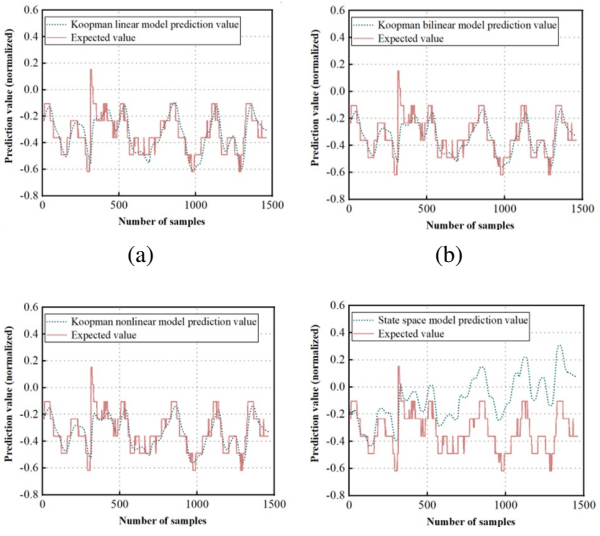
Table 2. Dimensions of five basis functions in the development of three Koopman models.

| Models | Dimensions of basis functions | | | | |
|---|---|---|---|---|---|
| | Monomial | Hermite | Gaussian | Fourier | Sparse Fourier |
| Linear | 10 | 10 | 2 | 124 | 27 |
| Bilinear | 10 | 10 | 2 | 124 | 27 |
| Nonlinear | 15 | 15 | 2 | 624 | 44 |

Table 3. Comparison of modeling performance across different models.

| Basis functions | Models | NRMSE | Improvement |
|---|---|---|---|
| Monomial | SSM | 0.1188 | – |
| | KLM | 0.0796 | 32.98% |
| | KBM | 0.0791 | 33.42% |
| | KNM | 0.0763 | 35.77% |
| Hermite | KLM | 0.0796 | 32.98% |
| | KBM | 0.0791 | 33.42% |
| | KNM | 0.0763 | 35.77% |
| Gaussian | KLM | 0.0815 | 31.39% |
| | KBM | 0.0806 | 32.32% |
| | KNM | 0.0801 | 32.58% |
| Fourier | KLM | 0.0807 | 32.07% |
| | KBM | 0.0795 | 33.08% |
| | KNM | 0.1401 | – |
| Sparse Fourier | KLM | 0.0838 | 29.46% |
| | KBM | 0.0825 | 30.56% |
| | KNM | 0.0812 | 31.65% |

Note: Improvement represents the improved values of Koopman models compared to that of the state space model (SSM). The definition in the following tables is the same as that in Table 3.




(a) (b) (c) (d)

Fig. 3. The prediction results of different models for the 2D soft robot: (a) KLM, (b) KBM, (c) KNM, (d) SSM.

Table 4. Quantitative comparison of prediction performance across different models.

| Models | NRMSE | Improvement |
|---|---|---|
| SSM | 0.3674 | — |
| KLM | 0.1119 | 69.54% |
| KBM | 0.1041 | 71.67% |
| KNM | 0.0966 | 73.71% |

### 3.2.2 Case study 2: Modeling and prediction of 3D soft robot

• **Modeling**

For the 3D soft robot, we choose the maximum degree $\rho = 2$, the time delay $d = 1$, $m = 7$ for both linear and bilinear models, and $m = 10$ for the nonlinear model. In addition, the dimensions of five basis functions in three Koopman models are summarized in Table 5.

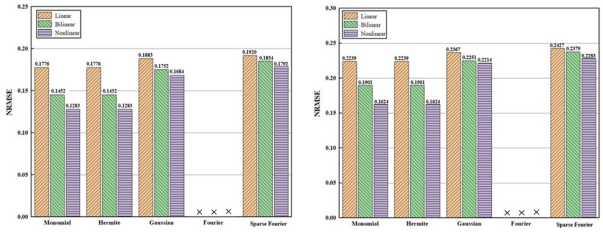
The modeling results of three Koopman models with five basis functions and the state space model for the 3D soft robot are quantitatively compared in Table 6 and Fig. 4(a). For the 3D soft robot, the NRMSE value is related to the NRMSE values for the X and Y axes which can be calculated by $\sqrt{\text{NRMSE(X)}^2 + \text{NRMSE(Y)}^2}$.

Table 5. Dimensions of five basis functions in three Koopman models.

| Models | Dimensions of basis functions | | | | |
|---|---|---|---|---|---|
| | Monomial | Hermite | Gaussian | Fourier | Sparse Fourier |
| Linear | 35 | 35 | 2 | 78124 | 119 |
| Bilinear | 35 | 35 | 2 | 78124 | 119 |
| Nonlinear | 65 | 65 | 2 | 9765624 | 230 |

Table 6. Comparison of modeling performance across different models.

| Basis functions | Models | NRMSE | Improvement |
|---|---|---|---|
| Monomial | SSM | 0.2052 | – |
| | KLM | 0.1776 | 13.94% |
| | KBM | 0.1452 | 29.24% |
| | KNM | 0.1283 | 37.48% |
| Hermite | KLM | 0.1776 | 13.94% |
| | KBM | 0.1452 | 29.24% |
| | KNM | 0.1283 | 37.48% |
| Gaussian | KLM | 0.1883 | 8.24% |
| | KBM | 0.1752 | 14.62% |
| | KNM | 0.1684 | 17.93% |
| Fourier | KLM | – | – |
| | KBM | – | – |
| | KNM | – | – |
| Sparse Fourier | KLM | 0.1920 | 6.43% |
| | KBM | 0.1854 | 9.65% |
| | KNM | 0.1792 | 12.67% |




(a) (b)

Fig. 4. The comparison of modeling and prediction results of three Koopman models with five basis functions (3D soft robot): (a) Modeling and (b) prediction. Note: 'x' in the Fourier case means no convergence.

• **Prediction**

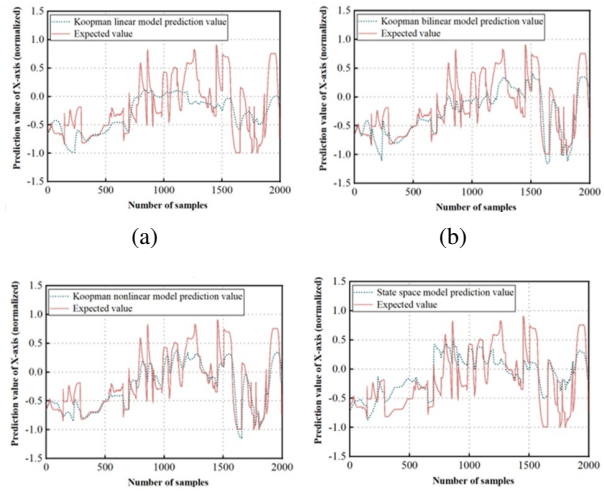
The comparison of the prediction results of three Koopman models with five basis functions for the 3D soft robot is shown in Fig. 4(b).

The prediction results ($X$ and $Y$ coordinates) of three developed Koopman models (Here, the basis function is chosen as the Monomial type) for the 3D soft robot are illustrated in Figs. 5 and 6.

Furthermore, the quantitative comparison of the modeling results of three constructed Koopman models and the state space model for the 3D soft robot is summarized in Table 7.

## 3.3. Discussion

The results in Subsection 3.2 clearly show that the Koopman models—linear, bilinear, and nonlinear—are significantly better than the traditional state-space model-




(a) (b)

(c) (d)

Fig. 5. The prediction results ($X$ coordinate) of different models for the 3D soft robot: (a) KLM, (b) KBM, (c) KNM, (d) SSM.

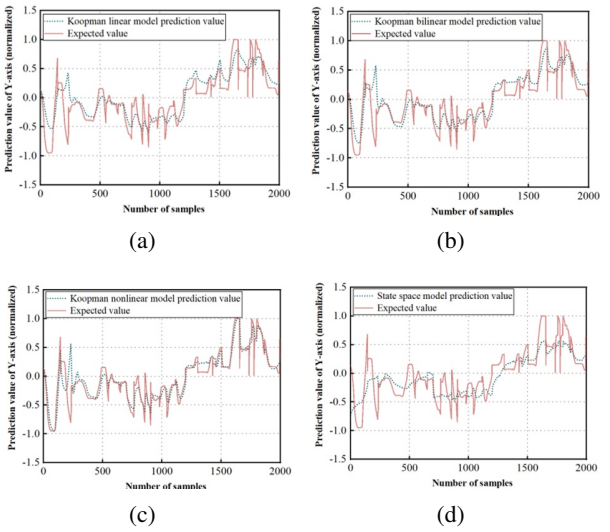



Fig. 6. The prediction results ($Y$ coordinate) of different models for the 3D soft robot: (a) KLM, (b) KBM, (c) KNM, (d) SSM.

Table 7. Quantitative comparison of prediction performance across different models.

| Models | NRMSE | Improvement |
|---|---|---|
| SSM | 0.2618 | — |
| KLM | 0.2239 | 14.47% |
| KBM | 0.1901 | 27.39% |
| KNM | 0.1624 | 37.96% |

ing for both 2D and 3D soft robots. The nonlinear Koopman model performs the best, followed by the bilinear and then the linear model. Using the monomial basis function as an example, we see that for the 2D soft robot, the overall modeling improvement for the three Koopman models is 32.98%, 33.42%, and 35.77%, respectively, and the prediction improvements are 69.54%, 71.67%, and 73.71%, respectively, over the existing state-space model. For the 3D soft robot, the modeling improvements are 13.94%, 29.24%, and 37.48%, and the prediction improvements are 8.92%, 26.01%, and 36.76%, respectively.

In our study, we've primarily used a trial-and-error approach to select basis functions for Koopman modeling, as there's a lack of comprehensive guidance on this topic. Our findings indicate that the Monomial basis function is the best for developing Koopman linear, bilinear, and nonlinear models for 2D and 3D soft robots. It's easy to implement and works well with polynomial vector fields, making it ideal for computing Koopman eigenfunctions. This function allows for linear state recovery without needing additional observables or solving extra optimization problems. Ongoing research is exploring the application of our modeling framework to composite control and dynamic prediction challenges. For instance, a wheeled mobile robot has employed a monomial basis function to build a Koopman linear model, which was then used with a sliding mode controller for effective control tasks, as validated in experiments [31]. Similarly, the monomial basis function has been effectively used to model complex and large-scale power systems, including wind turbines, and to predict their nonlinear trajectories [32]. In the future work, we plan to apply our Koopman models with various basis functions to improve model learning and control capabilities.

Table 8. Comparison of modeling performance with two kinds of basis functions.

| Basis functions | NRMSE | | | |
|---|---|---|---|---|
| | $\rho = 2$ | $\rho = 3$ | $\rho = 4$ | $\rho = 5$ |
| Monomial | 0.0796 | 0.7959 | 0.7957 | 0.7948 |
| Hermite | 0.0796 | 0.7959 | 0.7958 | 0.7950 |

Note: The value of $d$ is 1.

For the Hermite basis function, it has the same expression as the Monomial basis function as the maximum degree of basis function is less than four. Thus, the modeling and prediction results for the investigated soft robots obtained by using the Hermite basis function are equivalent to those by using the Monomial basis function in our study (the maximum degree of basis function in our studies is chosen as two). The use of the Hermite basis function and its inner product expression can greatly decrease the number of input features required for modeling nonlinear systems [33]. To do so, these compressed features can reduce the training time and complexity of the modeling process. In addition, the Hermite polynomials are simpler than the Gaussian and Fourier basis functions, which are best suited for problems where the data is normally distributed.

Note that, the modeling performance of Monomial as basis function is slightly better than the Hermite one when its maximum degree is larger or equal to four. For instance, Table 8 demonstrates the comparison of modeling results with two types of basis functions with different maximum degrees in the development of Koopman linear models for the 2D soft robot.

The Gaussian basis function, as mentioned in [34], is ideal for radial oscillatory dynamics like the Duffing oscillator. It has been effectively used in high-speed train modeling with the Koopman operator, outperforming other basis functions in terms of accuracy. However, it may not be the best fit for creating soft robot models in our research.

The Fourier basis function covers a wide span and is computationally demanding. Despite this, low-dimensional Fourier basis functions have shown good accuracy in modeling and prediction, especially for systems with oscillatory motion, as noted in [35]. This makes

them suitable for creating linear and bilinear models of 2D robots with positive outcomes due to their lower dimensions. Although the Fourier basis function might outperform Monomial, Hermite, and Gaussian functions, its convergence becomes challenging as the basis function's degree increases. For 3D soft robots, the Fourier basis function fails to converge when constructing Koopman linear, bilinear, and nonlinear models. Consequently, in Figs. 2(a) and 2(b), the Fourier cases exhibit a significantly larger NRMSE compared to the other models in the Koopman nonlinear model, demanding more computation and resulting in lower accuracy. In Figs. 4(a) and 4(b), convergence is not achieved.

The Sparse Fourier basis function is fast for computations and can reduce complexity, speeding up calculations, which is great for models with sparse data. However, it's not ideal for our models because our data is densely distributed to capture detailed information about soft robots. While the regular Fourier basis function struggles with convergence for 3D soft robot models, the Sparse Fourier basis function ensures convergence but may sacrifice accuracy in modeling and predictions.

## 4. CONCLUSIONS

In this paper, we have introduced a data-driven Koopman modeling framework designed to develop linear, bilinear, and nonlinear models for nonlinear dynamic systems by approximating the infinite-dimensional linear Koopman operator. We've assessed and discussed five basis functions to identify their most suitable applications. To test our methodology, it has been applied to model and predict the dynamics of both 2D and 3D soft robots, demonstrating its effectiveness. The detailed conclusions are further summarized as follows:

1) Koopman linear, bilinear, and nonlinear models for both 2D and 3D soft robots have higher performance than the existing state-space modeling approach. The established Koopman nonlinear model is better than the Koopman bilinear model, followed by the linear model. This conclusion is consistent no matter which basis function is chosen.
2) The Monomial basis function stands out as the best choice for creating Koopman models—linear, bilinear, and nonlinear—for both 2D and 3D soft robots. It's easy to implement and works well with polynomial vector fields. In our scenario, the Hermite basis function is equivalent to the Monomial because our basis functions are of a low degree, less than four. The Gaussian basis function is better for radial oscillatory dynamics, which makes it less suitable for our soft robot models. The low-dimensional Fourier basis function excels at modeling oscillatory systems but struggles with convergence as complexity increases. Despite this, it outperforms the Gaussian basis function and is nearly as effective as the Monomial and Hermite functions in linear and bilinear 2D soft robot modeling. The Sparse Fourier basis function reduces computational complexity and speeds up calculations, making it ideal for sparse data models. It ensures convergence, but this comes at the cost of some accuracy in modeling and prediction compared to the standard Fourier basis function.

We believe that our findings have the potential to be further exploited in a wide range of engineering and application fields that have to establish data-driven models without a physical model.

## CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

## REFERENCES

[1] Z. Liu, X. Yin, K. Peng, X. Wang, and Q. Chen, "Soft pneumatic actuators adapted in multiple environments: A novel fuzzy cascade strategy for the dynamics control with hysteresis compensation," *Mechatronics*, vol. 84, 102797, 2022.

[2] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315-318, 1931.

[3] S. E. Otto and C. W. Rowley, "Koopman operators for estimation and control of dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 59-87, 2021.

[4] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, "Robust tube-based model predictive control with Koopman operators," *Automatica*, vol. 137, 110114, 2022.

[5] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 948-961, 2020.

[6] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4369-4376, 2021.

[7] E. Kamenar, N. Ćrnjarić-Žic, D. Haggerty, S. Zelenika, E. W. Hawkes, and I. Mezić, "Prediction of the behavior of a pneumatic soft robot based on Koopman operator theory," *Proc. of 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, IEEE, pp. 1169-1173, 2020.

[8] G. Mamakoukas, M. L. Castano, X. Tan, and T. D. Murphey, "Derivative-based Koopman operators for real-time control of robotic systems," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2173-2192, 2021.

[9] M. L. Castaño, A. Hess, G. Mamakoukas, T. Gao, T. Murphey, and X. Tan, "Control-oriented modeling of soft robotic swimmer with Koopman operators," in *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, pp. 1679-1685, 2020.

[10] B. Chen, Z. Huang, R. Zhang, W. Liu, H. Li, J. Wang, Y. Fan, and J. Peng, "Data-driven Koopman model predictive control for optimal operation of high-speed trains," *IEEE Access*, vol. 9, pp. 82233-82248, 2021.

[11] N. Cassamo and J.-W. van Wingerden, "On the potential of reduced order models for wind farm control: A Koopman dynamic mode decomposition approach," *Energies*, vol. 13, no. 24, 6513, 2020.

[12] A. Surana, "Koopman operator framework for time series modeling and analysis," *Journal of Nonlinear Science*, vol. 30, no. 5, pp. 1973-2006, 2020.

[13] L. Shi, H. Teng, X. Kan, and K. Karydis, "A data-driven hierarchical control structure for systems with uncertainty," *Proc. of IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, pp. 57-63, 2020.

[14] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307-1346, 2015.

[15] M. Korda and I. Mezić, "On convergence of extended dynamic mode decomposition to the Koopman operator," *Journal of Nonlinear Science*, vol. 28, pp. 687-710, 2018.

[16] A. Salova, J. Emenheiser, A. Rupe, J. P. Crutchfield, and R. M. D'Souza, "Koopman operator and its approximations for systems with symmetries," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 9, 2019.

[17] S. Peitz and S. Klus, "Koopman operator-based model reduction for switched-system control of pdes," *Automatica*, vol. 106, pp. 184-191, 2019.

[18] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, 2017.

[19] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems," *Proc. of American Control Conference (ACC)*, IEEE, pp. 4832-4839, 2019.

[20] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[21] H. Shi and M. Q.-H. Meng, "Deep Koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700-7707, 2022.

[22] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Local Koopman operators for data-driven control of robotic systems," *Robotics: Science and Systems*, 2019.

[23] L. Shi and K. Karydis, "ACD-EDMD: Analytical construction for dictionaries of lifting functions in Koopman operator-based nonlinear robotic systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 906-913, 2021.

[24] I. Abraham, G. De La Torre, and T. D. Murphey, "Model-based control using Koopman operators," arXiv preprint arXiv:1709.01568, 2017.

[25] N. Parmar, H. H. Refai, and T. Runolfsson, "A survey on the methods and results of data-driven Koopman analysis in the visualization of dynamical systems," *IEEE Transactions on Big Data*, vol. 8, no. 3, pp. 723-738, 2020.

[26] S. Klus, P. Koltai, and C. Schütte, "On the numerical approximation of the perron-frobenius and Koopman operator," arXiv preprint arXiv:1512.05997, 2015.

[27] N. A. Kudryashov, "Generalized hermite polynomials for the burgers hierarchy and point vortices," *Chaos, Solitons & Fractals*, vol. 151, 111256, 2021.

[28] F. Soleymani and S. Zhu, "On a high-order gaussian radial basis function generated hermite finite difference method and its application," *Calcolo*, vol. 58, no. 4, p. 50, 2021.

[29] L. Zhang and S. Liu, "Iterative learning control for flexible manipulator using fourier basis function," *International Journal of Automation and Computing*, vol. 12, no. 6, pp. 639-647, 2015.

[30] L. Chen, "A note on the high-dimensional sparse fourier transform in the continuous setting," *Inverse Problems*, vol. 38, no. 3, 035008, 2022.

[31] C. Ren, H. Jiang, C. Li, W. Sun, and S. Ma, "Koopman-operator-based robust data-driven control for wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 1, pp. 461-472, 2022.

[32] Y. Xu, M. Netto, and L. Mili, "Propagating parameter uncertainty in power system nonlinear dynamic simulations using a Koopman operator-based surrogate model," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 3157-3160, 2022.

[33] C. Lowrie, "Classification using hermite basis functions," *Proc. of Fortieth Asilomar Conference on Signals, Systems and Computers*, IEEE, pp. 1143-1147, 2006.

[34] M. O. Williams, M. S. Hemati, S. T. Dawson, I. G. Kevrekidis, and C. W. Rowley, "Extending data-driven Koopman analysis to actuated systems," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 704-709, 2016.

[35] D. Bruder, *Towards a Universal Modeling and Control Framework for Soft Robots*, Ph.D. Dissertation, 2020.

**Lvpeng Han** received his B.S. degree in mechanical engineering from Yangtze University and an M.S. degree in mechanical engineering from Wuhan University of Technology, China, in 2021 and 2024, respectively. His research interests include nonlinear modeling and control of soft robots.

**Kerui Peng** received his B.S. degree in mechanical engineering from Henan University of Technology and an M.S. degree in mechanical engineering from Wuhan University of Technology, China, in 2020 and 2023, respectively. His research interests include advanced manufacturing technology, modeling and control of soft robots.

**Wangxing Chen** received his B.S. degree in mechanical engineering from North China Electric Power University and an M.S. degree in mechanical engineering from Wuhan University of Technology, China, in 2020 and 2024, resepctively. His research interests include nonlinear modeling and control of soft robots.

**Zhaobing Liu** received his B.S. degree in automation and an M.S. degree in control theory and control engineering from Northeastern University, China, and a Ph.D. degree in mechanical engineering from the University of Queensland, Australia, in 2006, 2008, and 2014, respectively. Now, he is an Associate Professor with Wuhan University of Technology. His current research interests include robotics and intelligent manufacturing.