



Sparse Identification of Lagrangian for Nonlinear Dynamical Systems Involving Dissipation Function

Yuan Tianjian

Department of Robotics,
 Graduate School of Engineering,
 Tohoku University,
 Sendai, Miyagi 980-8579, Japan
 e-mail: yuan.tianjian.q5@dc.tohoku.ac.jp

Adam Purnomo

Department of Robotics,
 Graduate School of Engineering,
 Tohoku University,
 Sendai, Miyagi 980-8579, Japan
 e-mail: adamsyammazaki@gmail.com

Mitsuhiro Hayashibe

Department of Robotics,
 Graduate School of Engineering,
 Tohoku University,
 Sendai, Miyagi 980-8579, Japan
 e-mail: hayashibe@tohoku.ac.jp

Accurately modeling nonlinear dynamic systems remains a significant challenge across numerous scientific fields, from physics to finance. Existing methods require prior knowledge of the system or are sensitive to noise, making them impractical to handle real-world tasks. Based on the sparse identification of nonlinear dynamics (SINDy) algorithm, our previous research proposed a variant of the SINDy algorithm. With the Lagrangian integrated, we named our method extended Lagrangian-SINDy (xL-SINDy). The xL-SINDy shows higher robustness than the competing method with a factor of 100, but it sacrifices the ability to deal with a nonconservative system. This paper introduces a new method that extends xL-SINDy's capabilities. We address its primary limitation by integrating the Rayleigh dissipation function into the xL-SINDy framework. Tests on four different pendulum systems show that our method maintains the same level of robustness as the xL-SINDy, but can identify nonconservative terms in the system. Making it better describe the real-world systems. While the proposed method is limited to specific types of dissipative forces that the Rayleigh dissipation can describe, the provided framework can be extended to include any form of the nonconservative terms. [DOI: 10.1115/1.4068081]

1 Introduction

More than three hundred years ago, Isaac Newton tried to figure out what made an apple fall off the tree; behind the phenomenon, he found gravity. That is one of our first attempts to find the system dynamics through the phenomenon. Since then, modeling has become one of the eternal topics in science research; even now, scientists are still trying to model the complex relationships between stars. However, there are too many dynamic systems in daily life, and it is impossible to model all of them manually. Still, knowing the dynamics might be important since it helps us to understand how the system's parts contribute to the movements. That makes the autonomous modeling method helpful for saving efforts. However, the noisy, complex environment makes real-world autonomous modeling more challenging than simulation, which is still an open topic [1–4].

There is a debate between complexity and parsimony in the quest for precision in modeling. Complex models offer detailed descriptions through complex equations but are often hard to interpret as explainable knowledge. A representation is the neural network model, which can fit the data with high accuracy, but it is hard to find analytical meaning from the network. Conversely, parsimonious models aim for simplicity and capture the system's essence with minimal components. As a representation, the equation of motion describes the system's movement and provides insights into the system's structure. Despite the difficulty of deriving such

models, contemporary research prefers parsimonious models for their analytical elegance [5–11].

Though studies in this area started more than a hundred years ago, applicable methods only began to appear after machine learning tools were developed [12–15]. In 2007, Lipson and co-worker explored merging machine learning into the autonomous modeling process [6]. In 2016, Brunton et al. proposed the game-changing sparse identification of nonlinear dynamics (SINDy) algorithm [7]. The machine learning method that removes the irrelevant terms and gives a sparse result, commonly called sparse regression, is the key that helps SINDy to lower the computation cost. That made SINDy to be the first applicable modeling algorithm. It was evaluated in many areas, from chemical kinetics to chaos theory [7,16–24].

With its proven performance, we choose it as the basement of this research. However, vanilla SINDy is sensitive to noise, so applying SINDy in the real world is challenging. To mitigate this issue, Brunton and co-workers proposed a more robust variant called SINDy with parallel and implicit expression (SINDy-PI) [25]. However, with the maximum capable Gaussian noise level as 10^{-3} , it was still not robust enough for real-world tasks. To make the algorithm even more robust, our group proposed the extended Lagrangian-SINDy (xL-SINDy) method, which changes the learning target from the equations of motion to Lagrangian, which describes the system dynamics in a simplified way [26,27]. Thanks to the simplicity of Lagrangian, the xL-SINDy can manage noise magnitude up to 10^{-1} , outperforming SINDy-PI. However, as a tradeoff, xL-SINDy cannot manage those systems with energy exchange with the outer environment, commonly called non-conservative systems. Since a typical energy exchange is the dissipative force and it is everywhere in the real world, most

This work was supported by the JSPS Grant-in-Aid for Scientific Research under Grant 24K00841.

Manuscript received October 8, 2024; final manuscript received February 25, 2025; published online March 27, 2025. Assoc. Editor: Wei-Che Tai.

dynamical systems in the real world are nonconservative. We need to extend the xL-SINDy to manage the nonconservative case before we apply it in the real world to solve the autonomous modeling challenge.

To better understand the relationship between SINDy, SINDy-PI, xL-SINDy, and our method, we can disassemble the whole recognition process into three parts:

- (1) The base algorithm: This determines the computational efficiency of the process.
- (2) The identification object: The object represents the system dynamics and works as the identification object; it affects the process robustness.
- (3) Formation of the regression loss: This affects the optimization criteria and brings the nonconservative capability of the process.

All four methods use the same base algorithm—the SINDy algorithm. For SINDy-PI, the identification object is replaced, providing an enhanced ability to handle greater robustness. xL-SINDy improved with the identification object and loss formation, with robustness surpassing SINDy-PI; however, it cannot handle nonconservative cases. In our research, we utilize the xL-SINDy framework but modify the loss to improve its capability, thereby retaining xL-SINDy’s superior robustness while overcoming the capability limitations. Figure 1 illustrates the comparison between SINDy and its variants.

In this paper, we combine a nonconservative expression called the Rayleigh dissipation function and the xL-SINDy, providing a highly robust and nonconservative capable solution for system identification and one step closer to solving real-world problems.

2 Methodology

This section introduces the vanilla SINDy and xL-SINDy and then discusses the modification to adopt a nonconservative system. A training efficiency test helps select the appropriate algorithm for the sparse regression task. Finally, the workflow of the whole algorithm is introduced, and a figure demonstration is given.

2.1 Vanilla Sparse Identification of Nonlinear Dynamics. As the basement of all other SINDy variants, vanilla SINDy enables efficient system dynamics modeling with a sparse regression procedure. Figure 2 explains the algorithm step-by-step. The simplest form separates the equations into the coefficients and system states’ functions. Then, the functions are stored in a vector, and the coefficients are stored in a matrix; the multiplication of these two can regenerate the equations. The next step assumes that the coefficients are unknown but that data from the system are given. In this case, the coefficient matrix would be initialized with random values, and a linear regression could help find the coefficients from the data. The regression loss is the error between the acceleration prediction and the true value. Finally, for the complete SINDy case, the coefficients and system states’ functions are unknown. The

initial function vector, also called the candidate library in SINDy, collects all possible combinations of the system states. The coefficient matrix would get randomly initialized, similar to the second step. The algorithm uses sparse regression to discover the relevant terms inside the huge function vector. It promotes most of the coefficients to zero, finding the dynamics from the data.

2.2 Extended Lagrangian Sparse Identification of Nonlinear Dynamics. Before introducing our modification to the xL-SINDy, it is necessary to introduce how xL-SINDy formulates the problem first. For vanilla SINDy, the equations of motion are disassembled into candidate functions and coefficients. In xL-SINDy, the Lagrangian is disassembled instead. Since Lagrangian does not give acceleration, it is necessary to use the Euler–Lagrange equation to obtain another regression loss: the system’s energy flow. The Euler–Lagrange equation describes a conservative situation whose dynamics can be fully described by the Lagrangian. The Euler–Lagrange equation can predict the energy flow in or out of the system; its error with the true value is the regression loss. With the regression loss obtained, we can perform the SINDy algorithm on the Lagrangian and fit the coefficients to the data. As the scalar describes the system dynamics, the Lagrangian is defined as the difference between the system’s kinetic and potential energy

$$\mathcal{L} = T - V \quad (1)$$

Using SINDy’s disassemble format, the Lagrangian can be separated into a candidate function vector Φ and a coefficients vector C . Note that vanilla SINDy has a separate expression for every system state, so a matrix is needed to store the coefficients. However, in xL-SINDy, the single expression of Lagrangian is enough to describe the dynamics. Thus, the candidate functions and coefficients are stored at a one-dimensional vector

$$\mathcal{L} = \Phi C, \Phi = [\phi_1 \ \phi_2 \ \dots \ \phi_p], C = [c_1 \ c_2 \ \dots \ c_p]^T \quad (2)$$

The Euler–Lagrange equation is defined as follows, and the q refers to the system states vector:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = 0 \quad (3)$$

Notice that each system state would have its partial derivative, if we expand the state vector, we would have individual equations for every state. In this paper, we would like to keep the system state in vector form for the simplicity of the expression. By substituting Eq. (2) into this equation, we can have

$$\frac{d}{dt} \left(\frac{\partial \Phi}{\partial \dot{q}} \right) C - \frac{\partial \Phi}{\partial q} C = 0 \quad (4)$$

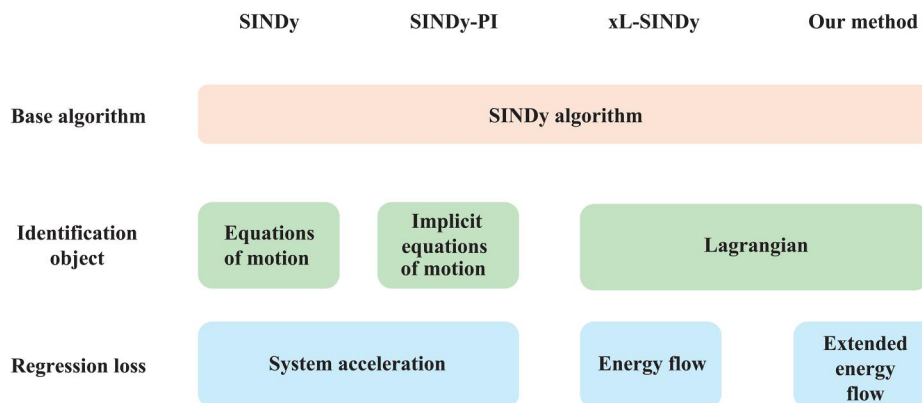


Fig. 1 Comparison between SINDy and its variants

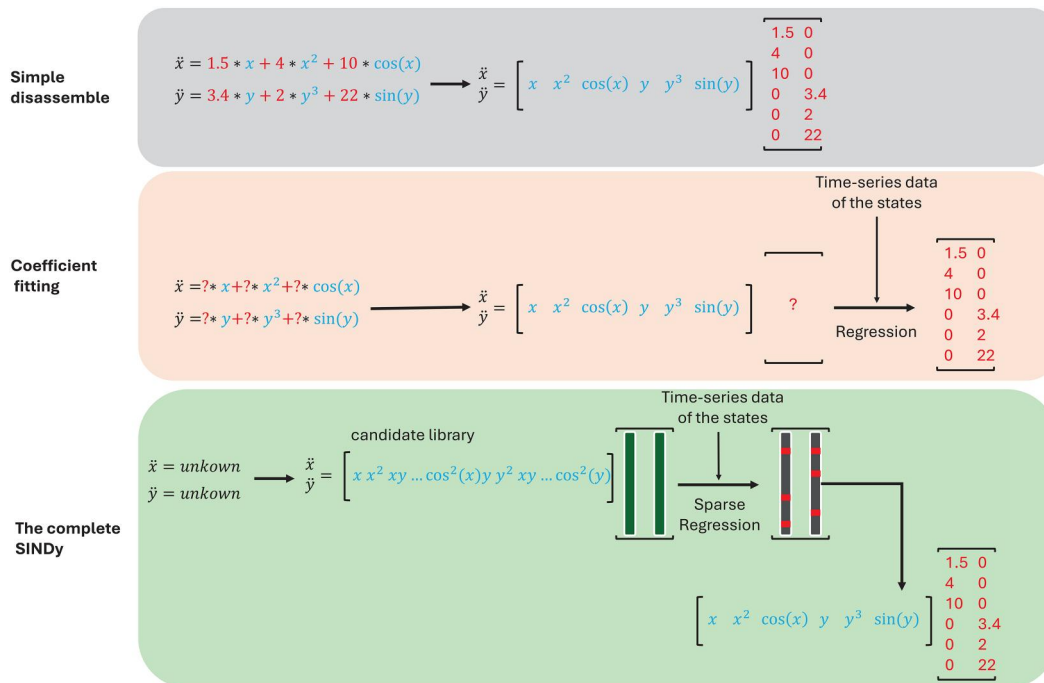


Fig. 2 A step-by-step demonstration of SINDy

The equation stands when the vector C is all zero, but that does not give any meaningful information about the system. And this zero-resolution trap should be avoided during regression training. Back to the derivation, Eq. (4) can be extended using the time derivative chain rule

$$\left(\frac{\partial}{\partial \dot{q}} \frac{\partial \Phi}{\partial \dot{q}} \dot{q} + \frac{\partial}{\partial q} \frac{\partial \Phi}{\partial \dot{q}} \dot{q} - \frac{\partial \Phi}{\partial q} \right) C = 0 \quad (5)$$

The regression process is to find a nonzero solution C' to make this equation stand, and its multiplication with the Φ gives the final learned Lagrangian. The defect of xL-SINDy comes from the zero on the right-hand side of Eq. (5), limiting it to only describe conservative systems.

2.3 Modified Euler–Lagrange Equation for Nonconservative Systems. To address the limitations of the previously described xL-SINDy method, which pertains specifically to conservative systems, we extend the capability of the Euler–Lagrange equation by adding a nonconservative term on the right-hand side. Specifically, we introduce the Rayleigh dissipation function model, which provides a universal solution for modeling various common types of dissipative force. It can generate different types of dissipative force by changing only one hyperparameter, being able to produce a large number of candidate functions in advance; it aligns with SINDy’s need for the candidate function vector construction. By adding the Rayleigh dissipation function terms into the candidate function vector, our algorithm can learn the system’s Lagrangian and its nonconservative parts at the same time. Although the Rayleigh dissipation function already contains the common types of dissipative force, it is necessary to admit that the Rayleigh dissipation function cannot cover all situations. Adding additional terms to the candidate function vector can address this issue.

To tackle different types of dissipation function, we incorporate the Rayleigh dissipation function directly into the Euler–Lagrange equation. Here, D stands for the Rayleigh dissipation function, and q_j is the j th generalized coordinate of the system

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = - \frac{\partial D}{\partial \dot{q}}, \quad D = \frac{1}{n+1} \sum_j e_j \dot{q}_j^{n+1} \quad (6)$$

The left side of this equation is the same as Eq. (3), with the Rayleigh dissipation function term added on the right-hand side. With a minus term as the system energy flow, the system loses energy due to dissipative force, which is the nonconservative case we want to describe.

Assuming we are working with a double pendulum system, \dot{q}_j refers to the velocity of the j th joint, and c_j refers to the coefficient of the j th joint’s dissipation function. The n refers to the order of the dissipation function. With such a function, we can implement different kinds of dissipative force into the system by changing n . For example, when $n = 0$, the function refers to a constant dissipation function like constant dissipative force, and when $n = 1$, it refers to a dissipation function that is linear to one of the system states like linear dissipative force. All different components can be easily added up to compose the whole dissipation function. In the experiment, the candidate function vector includes dissipative force created by the Rayleigh dissipation function for every system state with n ranging from 0 to 3, while a constant or linear type dissipative force is applied in the system.

To help understand how Rayleigh dissipation function works, here is a demonstration of how to use it to simulate linear dissipative force. Assume there is a system with the following Lagrangian and dissipative force. It describes a 1DOF object moving in a single dimension and experiencing linear dissipative force proportional to its velocity. Here, m is the object mass, \dot{x} is the object’s velocity, f is the dissipative force applied to the object, and b is the dissipative coefficient

$$\mathcal{L} = \frac{1}{2} m \dot{x}^2, \quad f = -b \dot{x} \quad (7)$$

According to the definition of Rayleigh dissipation function, the dissipation function should be set to $n = 1$

$$D = \frac{1}{2} b \dot{x} \quad (8)$$

We can verify this by substituting it into Eq. (6), which states that acceleration comes from the only external force: dissipative force.

The substitution meets the assumption in Eq. (7) and proves the capability of the Rayleigh dissipation function

$$\begin{aligned} \frac{d}{dt} \frac{\partial}{\partial \dot{x}} \left(\frac{1}{2} m \dot{x}^2 \right) - \frac{\partial}{\partial x} \left(\frac{1}{2} m \dot{x}^2 \right) &= - \frac{\partial}{\partial \dot{x}} \left(\frac{1}{2} b \dot{x}^2 \right) \\ \frac{d}{dt} (m \dot{x}) &= 0 - b \dot{x} \quad m \ddot{x} = -b \dot{x} = f \end{aligned} \quad (9)$$

After understanding how the Rayleigh dissipation function works, the next step is to add it to the xL-SINDy framework. For convenience in sparse regression, we need to divide the Lagrangian and the Rayleigh dissipation function into the candidates part and coefficient part as follows: $\mathcal{L} = \sum_{k=1}^p c_k \phi_k$, $D = \sum_{j=1}^h e_j \psi_j$.

Here, the ϕ_k and ψ_j refer to the Lagrangian and dissipation function candidates, respectively, and c_k and e_j refer to the coefficients of these two candidates' libraries.

With such assignments, Eq. (6) can be extended as follows:

$$\frac{d}{dt} \sum_{k=1}^p c_k \nabla_{\dot{q}} \phi_k - \sum_{k=1}^p c_k \nabla_q \phi_k = - \sum_{j=1}^h e_j \nabla_{\dot{q}} \psi_j \quad (10)$$

Then, the time derivative part can be further extended inside the candidate parts using the chain rule, and $\nabla_{\dot{q}}^T \nabla_{\dot{q}} \phi_k$ refers to two consecutive partial derivations on the ϕ_k with respect to the system states

$$\begin{aligned} \left(\sum_{k=1}^p c_k \nabla_{\dot{q}}^T \nabla_{\dot{q}} \phi_k \right) \ddot{q} + \left(\sum_{k=1}^p c_k \nabla_q^T \nabla_{\dot{q}} \phi_k \right) \dot{q} - \sum_{k=1}^p c_k \nabla_q \phi_k \\ = - \sum_{j=1}^h e_j \nabla_{\dot{q}} \psi_j \end{aligned} \quad (11)$$

To simplify the expression, we assign

$$\mathbf{M}_k = \nabla_{\dot{q}}^T \nabla_{\dot{q}} \phi_k, \quad \mathbf{N}_k = \nabla_q^T \nabla_{\dot{q}} \phi_k, \quad \mathbf{O}_k = \nabla_q \phi_k, \quad \mathbf{P}_j = \nabla_{\dot{q}} \psi_j \quad (12)$$

Then, the equation can be simplified as follows:

$$\sum_{k=1}^p c_k \left(\mathbf{M}_k \ddot{q} + \mathbf{N}_k \dot{q} - \mathbf{O}_k \right) = - \sum_{j=1}^h e_j \mathbf{P}_j \quad (13)$$

This demonstration mainly focuses on the passive scenario in which no active force is applied to the system, and no prior knowledge is assumed. With some modification, this method can also be applied to active systems and systems with partial prior knowledge. An active model can be obtained by including the dissipation function as part of the external force. The model with partial prior knowledge can be obtained by separating the dissipation function on the left-hand side of the equation.

Active case

$$\sum_{k=1}^p c_k \left(\mathbf{M}_k \ddot{q} + \mathbf{N}_k \dot{q} - \mathbf{O}_k \right) = \tau_{\text{ext}} - \sum_{j=1}^h e_j \mathbf{P}_j \quad (14)$$

The τ_{ext} refers to the known external force input to the system.

Partial prior knowledge case

$$- \mathbf{M}_l \ddot{q} - \mathbf{N}_l \dot{q} + \mathbf{O}_l = \sum_{\substack{k=1 \\ k \neq l}}^p c_k \left(\mathbf{M}_k \ddot{q} + \mathbf{N}_k \dot{q} - \mathbf{O}_k \right) + \sum_{j=1}^h e_j \mathbf{P}_j \quad (15)$$

Having some prior knowledge means knowing certain candidates in the expression and their coefficients. Here, \mathbf{M}_l , \mathbf{N}_l , \mathbf{O}_l refers to the components formed by the known candidates.

2.4 Sparse Regression Algorithm. The sparse regression training in the SINDy algorithm is challenging for its high sparsity compared to other common applications of sparse regression algorithms. For standard sparse regression cases, the ratio of irrelevant functions inside the whole vector, commonly named sparsity, is around 80% to 50%. In SINDy tasks, the sparsity is usually higher than 90%; such a sparse dataset is challenging to fit. To address this challenge, a robust, efficient, sparse regression algorithm is needed. In experiments, the proximal gradient descent (PGD) [28] methods perform well in high-sparsity sparse regression thanks to their robustness brought by the proximal operator. To find the optimal option, we tested three variants of PGD and vanilla PGD; the result is shown in Fig. 3. Although the PGD with line-search (PGD with Ls) [29] seems to converge with the least iterations, it takes longer for each iteration than all other methods. As a result, its actual time consumption sits between the accelerated proximal gradient descent (APGD) [30] and fast iterative shrinkage threshold algorithm (FISTA) [31], making the APGD the best choice for our task.

3 Results

In this section, we tested the algorithm's robustness on four different systems and then verified its generalizability with two extra tests. With a maximum capable noise level as 10^{-1} , the proposed method maintains the same robustness level as the xL-SINDy. In the generalizability test, the algorithm shows an accuracy of over 90% across different situations, proving its potential as a universal solution.

For ease of reading, the paper only compares the simulation curve between the true and obtained models. A table of all obtained expressions is in Table 1. In the single pendulum, θ refers to the angle of the pendulum; in cart pendulum, x refers to the displacement of the cart and θ refers to the angle of the pendulum; in the double pendulum, θ_1 refers to the angle of the first joint which is the joint that got fixed with the environment, θ_2 refers to the angle of the second joint which is the joint that connects the two bars; in the spherical pendulum, θ refers to the polar angle, and ϕ refers to the azimuthal angle.

We modified the training process with two special mechanisms besides the mentioned measures: two-phase training and scaling [32]. This accelerates and stabilizes the training process and does not affect robustness. With a weak connection with the main algorithm and page limitation, we skip the details of these mechanisms.

With Fig. 4 summarizing the whole algorithm, here is the training flow: First, select the Lagrangian and Rayleigh candidate functions, then randomly initialize the corresponding coefficients. By applying them to the Euler-Lagrange equation, we obtain the system's energy flow expression. Next, substitute the system states into Eq. (13), (14), or (15) to predict the system's energy flow. Using the error

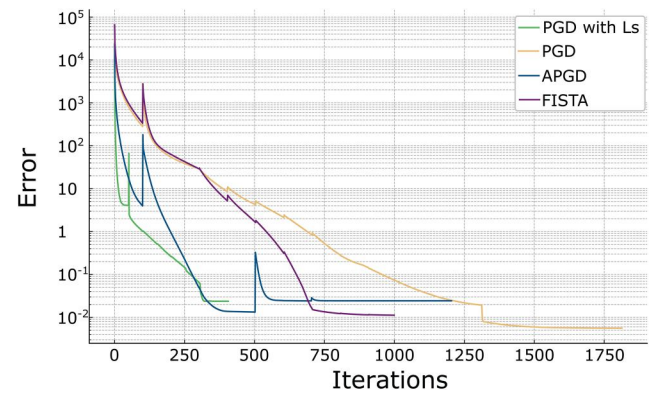


Fig. 3 Training efficiency comparison, the error is the mean square error between the angle prediction from the obtained model and the true angle value

Table 1 Obtained Lagrangian and Rayleigh dissipation functions for different systems under various noise levels

System	Noise	Obtained Lagrangian	Obtained dissipation function
Single pendulum	True	$0.500\dot{\theta}^2 + 9.81 \cos \theta$	$0.090\dot{\theta}^2$
	0	$0.500\dot{\theta}^2 + 9.81 \cos \theta$	$0.090\dot{\theta}^2$
	10^{-3}	$0.500\dot{\theta}^2 + 9.61 \cos \theta$	$0.090\dot{\theta}^2$
	2×10^{-2}	$0.500\dot{\theta}^2 + 9.87 \cos \theta$	$0.090\dot{\theta}^2$
	6×10^{-2}	$0.500\dot{\theta}^2 + 9.49 \cos \theta$	$0.090\dot{\theta}^2$
	10^{-1}	$0.500\dot{\theta}^2 + 9.79 \cos \theta$	$0.090\dot{\theta}^2$
Cart pendulum	True	$0.250\dot{\theta}^2 + 0.750\dot{x}^2 + 0.500\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.250\dot{\theta}^2 + 0.250\dot{x}^2$
	0	$0.250\dot{\theta}^2 + 0.740\dot{x}^2 + 0.495\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.250\dot{\theta}^2 + 0.250\dot{x}^2$
	10^{-4}	$0.250\dot{\theta}^2 + 0.736\dot{x}^2 + 0.493\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.250\dot{\theta}^2 + 0.251\dot{x}^2$
	10^{-3}	$0.245\dot{\theta}^2 + 0.690\dot{x}^2 + 0.461\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.249\dot{\theta}^2 + 0.251\dot{x}^2$
	2×10^{-2}	$0.245\dot{\theta}^2 + 0.678\dot{x}^2 + 0.453\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.252\dot{\theta}^2 + 0.235\dot{x}^2$
	6×10^{-2}	$0.245\dot{\theta}^2 + 0.649\dot{x}^2 + 0.436\dot{x}\dot{\theta} \cos \theta + 4.905 \cos \theta$	$0.241\dot{\theta}^2 + 0.192\dot{x}^2$
Double pendulum	True	$\dot{\theta}_1^2 + 0.5\dot{\theta}_2^2 + 19.62 \cos \theta_1 + 9.81 \cos \theta_2 + \dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + \dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.250\dot{\theta}_1^2 + 0.250\dot{\theta}_2^2$
	0	$0.999\dot{\theta}_1^2 + 0.500\dot{\theta}_2^2 + 19.620 \cos \theta_1 + 9.794 \cos \theta_2 + 0.999\dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + 1.000\dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.241\dot{\theta}_1^2 + 0.244\dot{\theta}_2^2$
	10^{-3}	$0.998\dot{\theta}_1^2 + 0.498\dot{\theta}_2^2 + 19.620 \cos \theta_1 + 9.752 \cos \theta_2 + 0.998\dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + 0.998\dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.251\dot{\theta}_1^2 + 0.249\dot{\theta}_2^2$
	2×10^{-2}	$0.998\dot{\theta}_1^2 + 0.498\dot{\theta}_2^2 + 19.620 \cos \theta_1 + 9.764 \cos \theta_2 + 0.998\dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + 0.998\dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.234\dot{\theta}_1^2 + 0.240\dot{\theta}_2^2$
	6×10^{-2}	$0.999\dot{\theta}_1^2 + 0.494\dot{\theta}_2^2 + 19.620 \cos \theta_1 + 9.755 \cos \theta_2 + 0.995\dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + 0.994\dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.219\dot{\theta}_1^2 + 0.223\dot{\theta}_2^2$
	10^{-1}	$0.988\dot{\theta}_1^2 + 0.488\dot{\theta}_2^2 + 19.620 \cos \theta_1 + 9.608 \cos \theta_2 + 0.973\dot{\theta}_1\dot{\theta}_2 \sin \theta_1 \sin \theta_2 + 0.983\dot{\theta}_1\dot{\theta}_2 \cos \theta_1 \cos \theta_2$	$0.198\dot{\theta}_1^2 + 0.211\dot{\theta}_2^2$
Spherical pendulum	True	$0.50\dot{\theta}^2 + 0.50\dot{\phi}^2 \sin^2 \theta + 9.81 \cos \theta$	$0.050\dot{\theta}^2 + 0.050\dot{\phi}^2$
	0	$0.50\dot{\theta}^2 + 0.50\dot{\phi}^2 \sin^2 \theta + 9.81 \cos \theta$	$0.050\dot{\theta}^2 + 0.050\dot{\phi}^2$
	10^{-3}	$0.51\dot{\theta}^2 + 0.511\dot{\phi}^2 \sin^2 \theta + 9.79 \cos \theta$	$0.053\dot{\theta}^2 + 0.049\dot{\phi}^2$
	2×10^{-2}	$0.54\dot{\theta}^2 + 0.44\dot{\phi}^2 \sin^2 \theta + 9.81 \cos \theta$	$0.046\dot{\theta}^2 + 0.048\dot{\phi}^2$
	6×10^{-2}	$0.48\dot{\theta}^2 + 0.48\dot{\phi}^2 \sin^2 \theta + 9.81 \cos \theta$	$0.028\dot{\theta}^2 + 0.042\dot{\phi}^2$

between the prediction and the true value, update the coefficients and fit them to the data, which gives the final Lagrangian and Rayleigh dissipation functions. We also provide some visualizations of the libraries in Figs. 10–13; these figures are provided in the Appendix.

In the results, we noticed that the coefficients for the dissipative function tend to decline with the growing noise magnitude, a possible explanation would be the dissipative force’s effect is gradually being covered with the growing noise, making it harder to recognize the dissipative part under high noise level.

3.1 Experiment Setup. To evaluate the effectiveness of our method, we selected single pendulum, cart pendulum, double pendulum, and spherical pendulum systems as subjects for our experimentation; the detailed configurations of these systems are

defined in our previous work [27] and shown in Fig. 5. These choices were made due to their relatively simple but chaotic behavior, making them ideal candidates for comparison with SINDy-PI. The dynamics governing these systems are defined by their respective Lagrangian, as presented in Table 1. To include dissipative part in the system, a linear dissipative force is applied to all four systems, as the dissipation function shows in Table 1. The whole process is based on simulation, and the data sample frequency is 50 Hz.

To evaluate the robustness of the proposed method, we added Gaussian noise into the generated data. We aligned our noise magnitude settings with those of xL-SINDy for comparison convenience, selecting values of 10^{-4} , 10^{-3} , 2×10^{-2} , 6×10^{-2} , and 10^{-1} . The tests reveal that the proposed method maintains the same robustness level as the xL-SINDy and outperforms SINDy-PI but with the extra capability to model nonconservative systems.

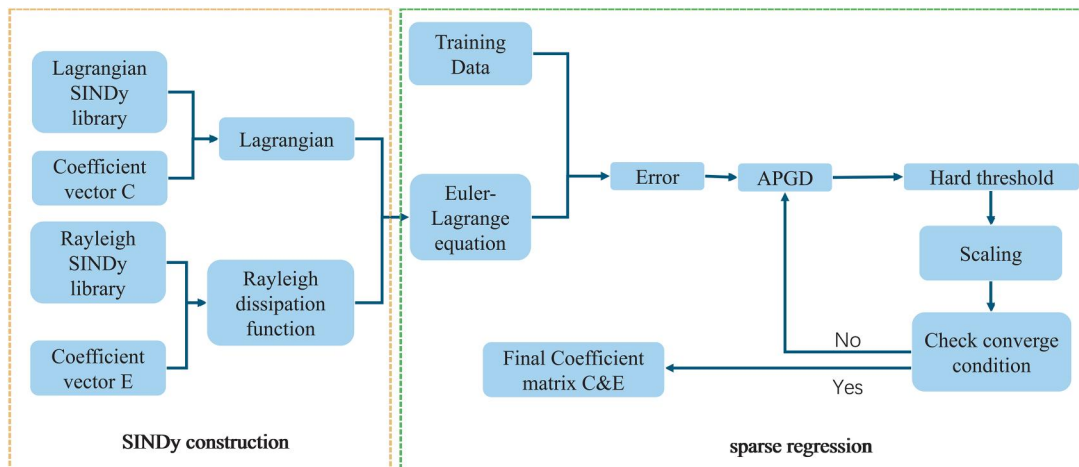


Fig. 4 The workflow of the proposed algorithm

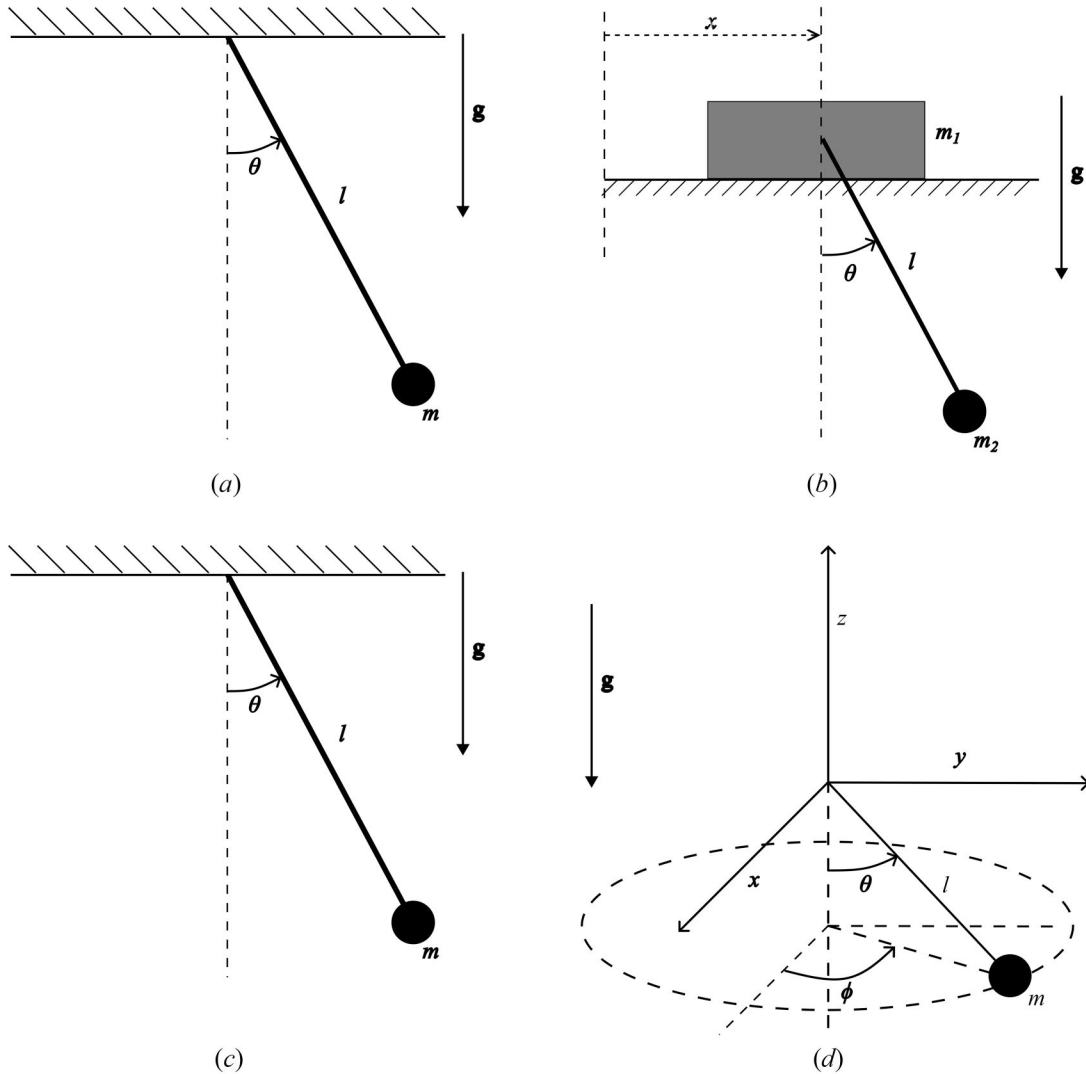


Fig. 5 Overview of different pendulum structures used in experiment. (a) The structure of the single pendulum. In the experiment, $l = 1$ m and $m = 1$ kg. (b) The structure of the cart pendulum. In the experiment, $l = 1$ m and $m = 1$ kg. (c) The structure of the double pendulum. In the experiment, $l_1 = l_2 = 1$ m and $m_1 = m_2 = 1$ kg. (d) The structure of the spherical pendulum. In the experiment, $l = 1$ m and $m = 1$ kg.

To show that the algorithm can perform stably with different pendulum systems, we also tested the algorithm under the same model type but with different physical parameters and dissipative force types.

For the sake of readability and to aid in interpreting results, all Lagrangians in this study have been scaled relative to the first coefficient. This approach is valid, as scaling the Lagrangian by an arbitrary constant does not alter the system dynamics. For the same reason, systems under different noise levels are started at the same initial position in the figures below, though in real experiments the position is randomly initialized. In the figures, the first 5 s would be the training data with noise and the obtained model's data, while the rest would be the validation data that started from a different initial position and the obtained model's data. By showing both the training and validation comparison to the obtained model, it is proved that it is indeed a universal model that fully describes the system's dynamics.

3.2 Single Pendulum. For the single pendulum system, we attempt to recover the system dynamics with the Rayleigh dissipation function model. In sample generation, the pendulum is randomly initialized in the range $[-\pi, \pi]$ for the angle and zero for the velocity. The candidate function library includes polynomials

and trigonometric functions up to second order for the Lagrangian, and dissipation functions for n ranging from 0 to 3, eventually building a library containing 17 functions, with three true functions inside the library (two from the Lagrangian and one from the dissipation function); the task sparsity is 17.64%. Figure 6 gives an intuitive simulation based on the result of the Rayleigh dissipation function incorporated method. The robustness of the algorithm is also proven. Although not shown here, for the conservative system, the coefficient shifting is less prominent along with the growing noise; the same magnitude of noise has a greater influence on nonconservative system. This effect stems from a larger relative error when the oscillation magnitude decreases at the late stage of nonconservative systems. Overall, the positive result in this simple model proves the algorithm's viability, giving confidence for tests on complex systems.

3.3 Cart Pendulum. The angle θ of the pendulum is randomly initialized at the range $[-\pi, \pi]$, and the cart position is zero, both with zero velocity. The candidate function library includes polynomials and trigonometric functions up to fourth order for the Lagrangian, and dissipation functions with n ranging from 0 to 3, eventually building a library containing 61 functions, with six true functions inside the library (four from the Lagrangian and two from the

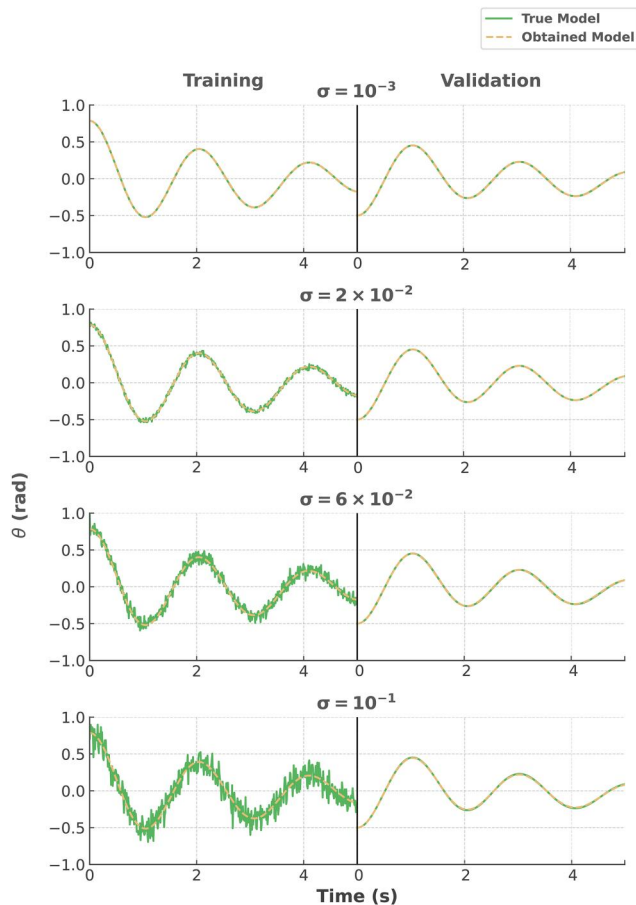


Fig. 6 Single pendulum simulation results (Rayleigh model)

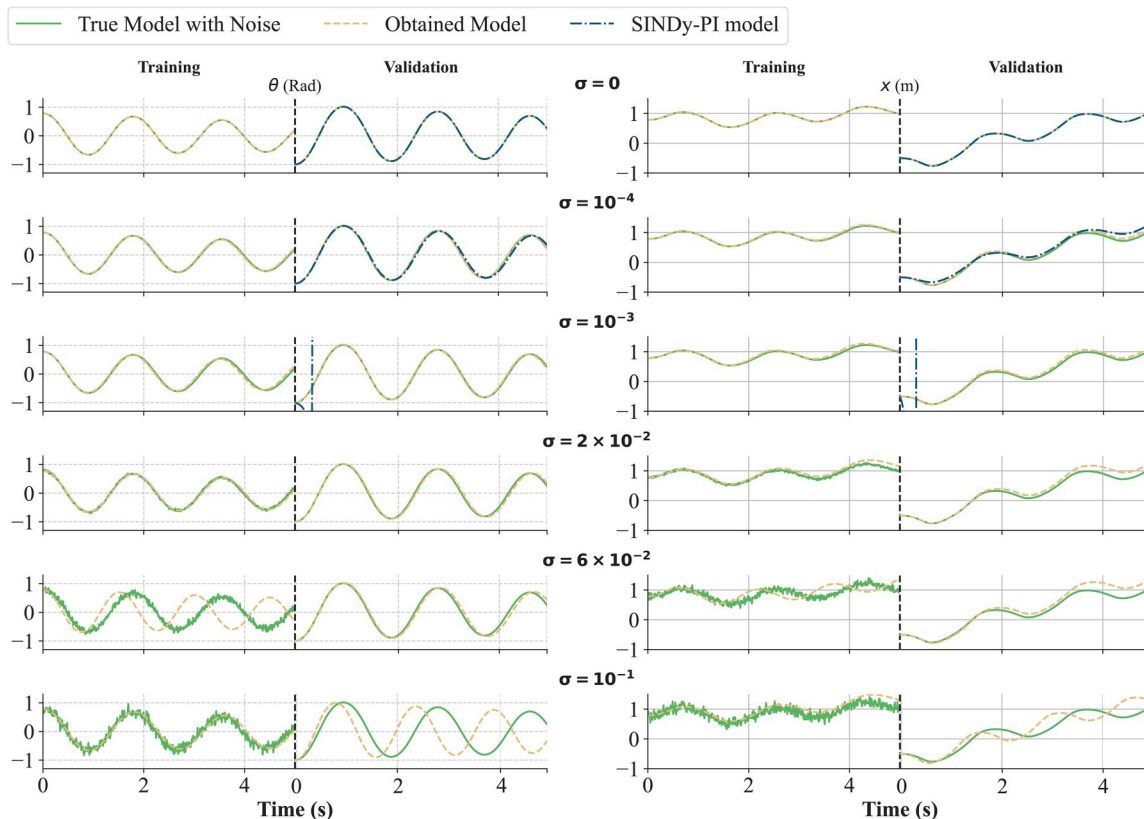


Fig. 7 Cart pendulum simulation result based on the model obtained by our method and SINDy-PI, the training data consists of 20 initial conditions, each lasting for 5 s. The validation set is the data for 5 s with a different initial condition.

dissipation function); the task sparsity is 9.84%. The simulation result is shown in Fig. 7. As the results show, our method can find the correct candidates for noise levels up to 10^{-1} , although with some coefficients shifting, the simulation curve of our obtained can still follow the true model.

We also performed a comparative analysis using the well-known SINDy-PI method to validate our proposed algorithm further. In Fig. 7, the dotted lines give the results of SINDy-PI. As the results indicate, SINDy-PI method is more sensitive to noise. Even though it is claimed that SINDy-PI can handle noise magnitudes up to 10^{-3} , the actual robustness of the algorithm is also affected by the system complexity. The involvement of dissipative force significantly adds complexity, causing SINDy-PI's performance degradation. In our cart pendulum test, SINDy-PI can handle only up to 10^{-4} noise magnitude. In contrast, our proposed method can accommodate noise magnitudes up to 10^{-1} , matching the robustness of the original xL-SINDy algorithm.

3.4 Double Pendulum. The pendulum is randomly initialized for sampling, both joints' angles are in the range $[-\pi, \pi]$, and velocities are zero. The candidate function library has 95 candidates, with eight true functions in the library (six for the Lagrangian and two for the dissipation function); the task sparsity is 8.42%. Testing with the double pendulum system in Table 1, Fig. 8 offers an intuitive simulation-based comparison between the model generated by our algorithm, the model generated by SINDy-PI, and the true model. Our algorithm effectively recovers the correct candidate terms up to a noise magnitude of 10^{-1} , though there is a noticeable shift in coefficients when the noise level reaches 6×10^{-2} . This could be attributed to the relative noise level issue discussed in Sec. 3.2. Still, the algorithm exhibits 10^3 times higher robustness than SINDy-PI.

Like the cart pendulum case, SINDy-PI could only handle noise level up to 10^{-4} . For higher noise level, the SINDy-PI can only discover part of the true model or even fail to capture any essence of

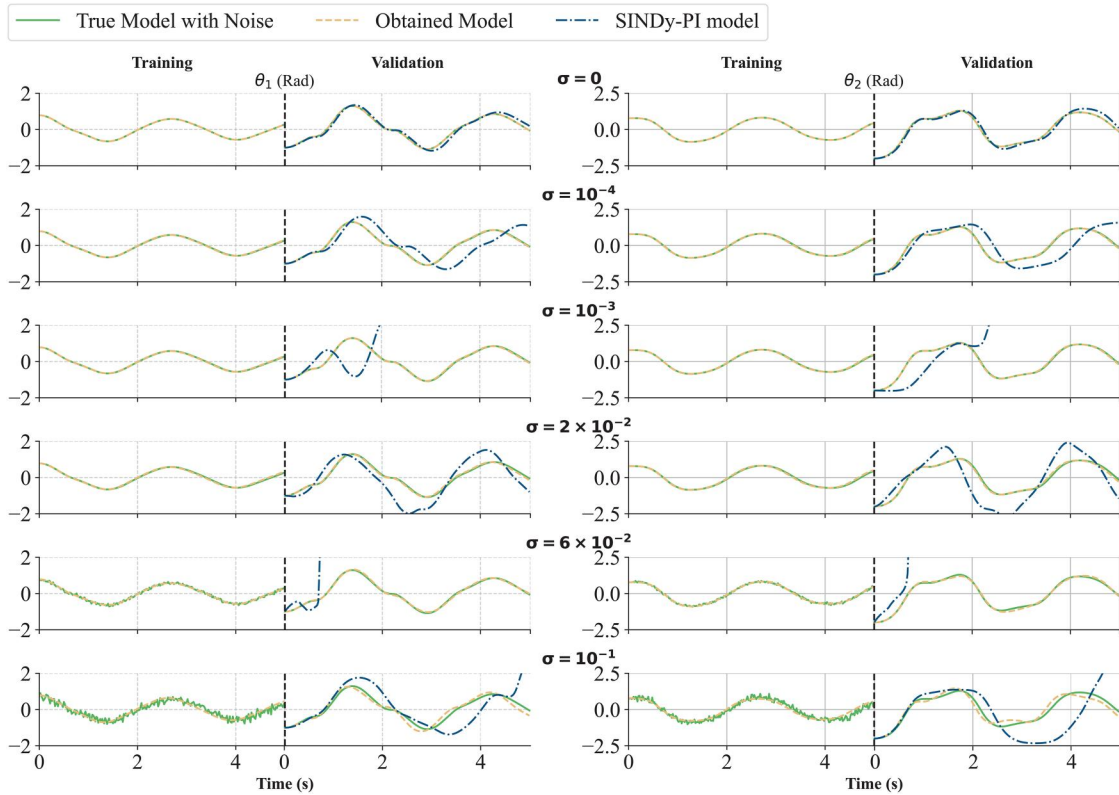


Fig. 8 Double pendulum simulation result based on the model obtained by our method and SINDy-PI, the training data consists of 20 initial conditions, each lasting for 5 s. The validation set is the data for 5 s with a different initial condition.

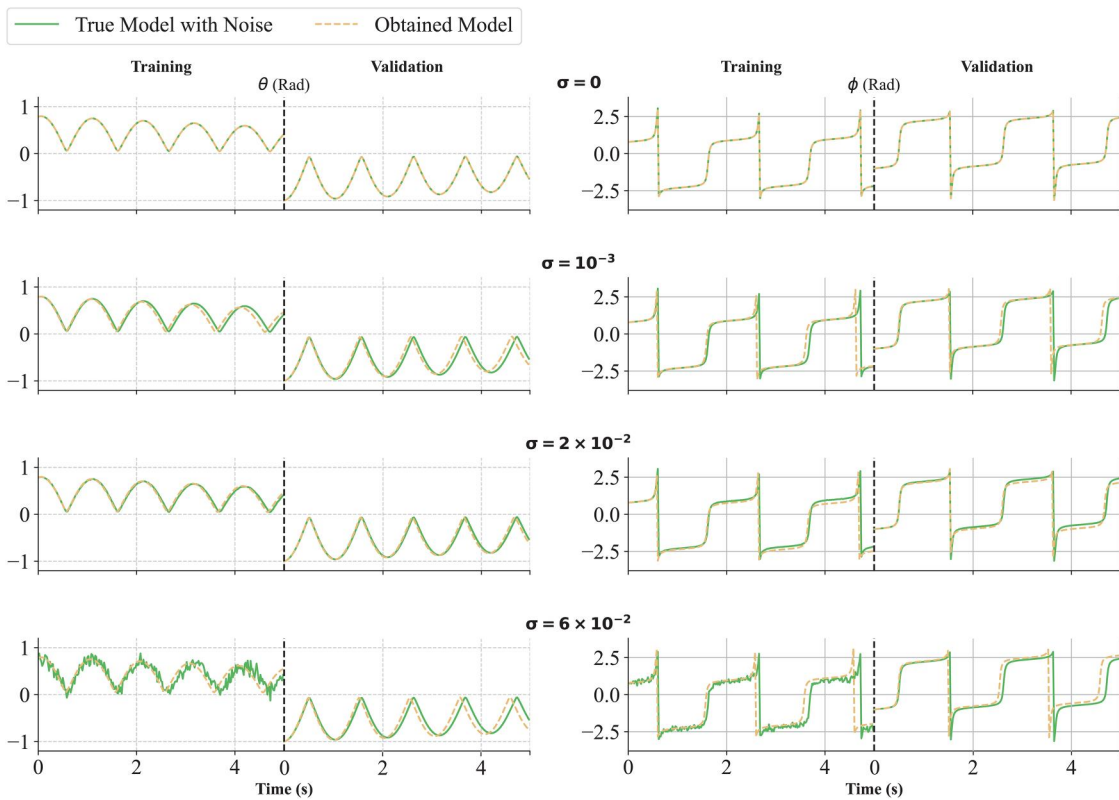


Fig. 9 Simulation result of the spherical pendulum

Table 2 Accuracy on the double pendulum with different physical conditions

$b_1 = 0.2, b_2 = 0.2$			
	$l_1 = 0.5,$ $b = 0.5$	$l_1 = 1,$ $b_2 = 1$	$l_1 = 1.5,$ $b_2 = 1.5$
$m_1 = 1.5,$ $m_2 = 1.5$	94.57%	93.12%	96.72%
$m_1 = 1,$ $m_2 = 1$	93.74%	93.97%	93.38%
$m_1 = 0.5,$ $m_2 = 0.5$	94.19%	94.53%	94.32%
$b_1 = 0.05, b_2 = 0.05$			
	$l_1 = 0.5,$ $l_2 = 0.5$	$l_1 = 1,$ $l_2 = 1$	$l_1 = 1.5,$ $l_2 = 1.5$
$m_1 = 1.5,$ $m_2 = 1.5$	93.64%	91.03%	92.41%
$m_1 = 1,$ $m_2 = 1$	94.37%	92.35%	94.06%
$m_1 = 0.5,$ $m_2 = 0.5$	94.82%	93.85%	92.44%

Table 3 Model accuracy under different dissipative coefficients

(a) Viscous dissipative force					
b_2	acc	b_1	0.05	0.1	0.5
0.05			92.35%	95.57%	92.53%
0.1			95.70%	95.57%	93.57%
0.5			96.20%	94.52%	94.20%
(b) Constant dissipative force					
b_2	acc	b_1	0.05	0.1	0.5
0.05			98.91%	95.92%	98.59%
0.1			98.12%	98.33%	98.18%
0.5			96.02%	95.16%	96.58%

it. In the meantime, our method can still find the correct candidates and follow the true model's curve at the noise level of 10^{-1} , further proving the robustness of our method.

3.5 Spherical Pendulum. Finally, we also tested our method using a spherical pendulum mentioned in Table 1. With a polar coordinate definition, the pendulum's polar angle θ is initialized in the range $[\pi/3, \pi]$, while the initial azimuthal angle is zero, and the angle velocities are both zero. The candidate function library has 65, with five true functions in the library (three for the Lagrangian and two for the dissipation function); the task sparsity is 7.69%.

The test results are shown in Fig. 9; compared with the other three cases, the spherical pendulum turns out to be more challenging to recognize. Our method can only handle noise levels up to 6×10^{-2} , as SINDy-PI could not give any valid result even for cases without noise.

3.6 Generalizability Test. To evaluate the algorithm's generalizability, the algorithm is also tested under varying physical parameters. To be more specific, we utilize the double pendulum system and change its mass, length, and damping coefficients. Table 2 provides a visual representation of the model's accuracy through a heatmap under a noise level of 10^{-3} ; the deeper color refers to a higher accuracy. The accuracy is calculated using the mean square error of the coefficients between the identified model and the true model. Here, l_1, l_2 refer to the length of the pendulums,

m_1, m_2 refer to the masses, and b_1, b_2 refer to the dissipative coefficients. Across various physical parameters, our algorithm consistently recovered system dynamics with an accuracy exceeding 90%, affirming its promising applicability in diverse settings. Besides, model performance under different dissipative force types and magnitudes is also tested. The test contains constant and viscous dissipative force, with coefficients ranging from 0.05 to 0.5. With results shown in Table 3, the overall accuracy stays above 90%, showing the algorithm's capability. The result also shows that the algorithm performs better under constant dissipative force. Overall, the proposed method generalizes different system settings and dissipative force, proving its potential as a universal solution.

4 Conclusion and Discussion

This paper presents an algorithm for discovering the Lagrangian models of nonconservative systems using noisy data. Extending the existing algorithm to the on-conservative domain is critical for future real-world applications. Our method improves noise robustness compared with the existing SINDy-PI method.

While the xL-SINDy offered a robust approach to identifying the conservative systems, its inability to account for nonconservative systems was a notable drawback. The proposed method fills this gap effectively, offering a more generalized modeling approach. The Rayleigh model we proposed has excellent capabilities for different situations of dissipative force. It is necessary to admit that the proposed method can only handle dissipative forces included by the Rayleigh dissipation function. However, such a capability can be extended by manually adding additional terms to the dissipation function library. With these extra terms, other nonconservative nonlinear forces like Coulomb friction or viscous friction can also be detected in the model.

A current limitation of our approach is the recognition efficiency and accuracy when the system becomes complicated. A robustness degradation already appears in the spherical pendulum test; such degradation would be more evident for more complex systems. Some pre/postdata processing techniques are under consideration to further improve the robustness. In light of recent advancements, it is crucial to acknowledge the Ensemble-SINDy methodology introduced by Fasel et al. [33], which has exhibited significant gains in robustness and accuracy by incorporating ensemble techniques. This is a postprocess of the results, and it is particularly appealing for our work, as the ensemble method is independent of the core SINDy algorithm, enabling seamless integration with any of the variants of SINDy. Moreover, approaches like derivative-based SINDy [34], which employ denoising techniques before regression as a data preprocessing, offer another compelling avenue for enhancing the robustness of our model. Unlike our method and SINDy-PI, these two methods did not modify the SINDy process. Therefore, they are excluded from the comparison in the experiment part. Still, their performance is appealing, and integrating them might be our next step.

Our method offers a robust approach for the sparse identification of nonlinear dynamic systems. Its ability to incorporate dissipation functions makes it closer to real-world applications, setting new stage for future research in this area.

Funding Data

- JSPS Grant-in-Aid for Scientific Research (Grant No. 24K00841; Funder ID: 10.13039/501100001691).

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Nomenclature

b = damping coefficient
 b_1, b_2 = the damping coefficient of the first and second joint of double pendulum
 C = coefficient vector
 c_k, c_j = coefficients of the system state functions
 D = Rayleigh dissipation function
 f = damping force
 \mathcal{L} = Lagrangian
 l_1, l_2 = the length of the first and second bar of double pendulum
 m = object mass
 m_1, m_2 = the mass of the first and second end of double pendulum
 $\mathbf{M}_k, \mathbf{N}_k, \mathbf{O}_k, \mathbf{P}_j$ = simplification signs for the equation
 $\mathbf{M}_j, \mathbf{N}_j, \mathbf{O}_j$ = simplification signs for the known part of the equation
 n = the hyperparameter of Rayleigh dissipation function
 \mathbf{q} = system state
 T = kinetic energy

V = potential energy
 x, \dot{x}, \ddot{x} = object position, velocity, and acceleration
 ∇ = differential operator
 $\theta, \dot{\theta}, \ddot{\theta}$ = object angle, angular velocity, and angular acceleration
 $\theta_1, \dot{\theta}_1$ = the angle and angular velocity of double pendulum's first joint
 $\theta_2, \dot{\theta}_2$ = the angle and angular velocity of double pendulum's second joint
 σ = Gaussian noise magnitude
 τ_{ext} = external force to the system
 $\dot{\phi}$ = the azimuthal angular velocity of spherical pendulum
 ϕ_k = system state functions
 Φ = function vector
 ψ_j = Rayleigh dissipation functions

Appendix: Candidate Library Visualization

In this appendix, we add visualization plots for the candidate library used in the experiment for ease of understanding, the highlighted expressions are the true terms for the system dynamics.

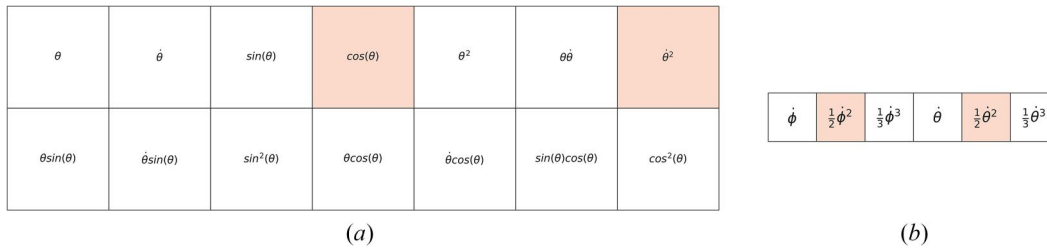


Fig. 10 Candidate library visualization of the single pendulum: (a) the Lagrangian library and (b) the Rayleigh dissipation function library

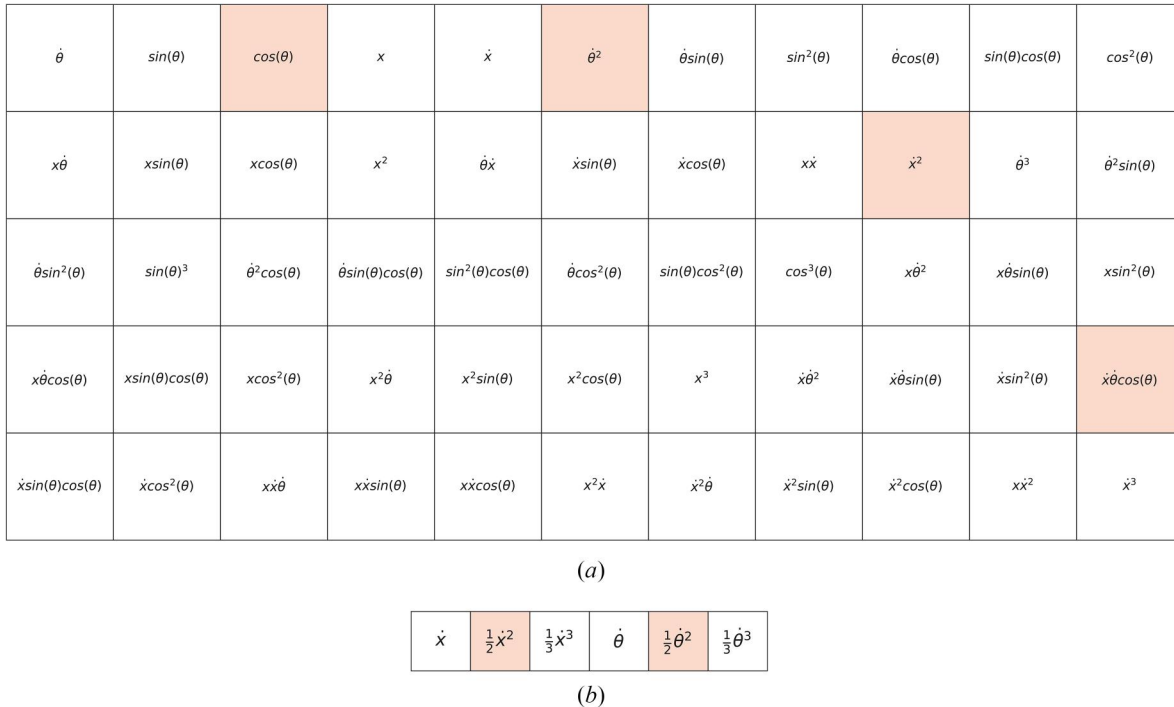


Fig. 11 Candidate library visualization of the cart pendulum: (a) the Lagrangian library and (b) the Rayleigh dissipation function library

$\dot{\theta}_1$	$\dot{\theta}_2$	$\dot{\theta}_1^2$	$\dot{\theta}_1\dot{\theta}_2$	$\dot{\theta}_2^2$	$\sin(\theta_1)$	$\sin(\theta_2)$	$\cos(\theta_1)$	$\cos(\theta_2)$	$\sin^2(\theta_1)$
$\sin(\theta_1)\sin(\theta_2)$	$\sin^2(\theta_2)$	$\sin(\theta_1)\cos(\theta_1)$	$\sin(\theta_2)\cos(\theta_1)$	$\cos^2(\theta_1)$	$\sin(\theta_1)\cos(\theta_2)$	$\sin(\theta_2)\cos(\theta_2)$	$\cos(\theta_1)\cos(\theta_2)$	$\cos^2(\theta_2)$	$\dot{\theta}_1\sin(\theta_1)$
$\dot{\theta}_1\sin(\theta_2)$	$\dot{\theta}_1\cos(\theta_1)$	$\dot{\theta}_1\cos(\theta_2)$	$\dot{\theta}_1\sin^2(\theta_1)$	$\dot{\theta}_1\sin(\theta_1)\sin(\theta_2)$	$\dot{\theta}_1\sin^2(\theta_2)$	$\dot{\theta}_1\sin(\theta_1)\cos(\theta_1)$	$\dot{\theta}_1\sin(\theta_2)\cos(\theta_1)$	$\dot{\theta}_1\cos^2(\theta_1)$	$\dot{\theta}_1\sin(\theta_1)\cos(\theta_2)$
$\dot{\theta}_1\sin(\theta_2)\cos(\theta_2)$	$\dot{\theta}_1\cos(\theta_1)\cos(\theta_2)$	$\dot{\theta}_1\cos^2(\theta_2)$	$\dot{\theta}_2\sin(\theta_1)$	$\dot{\theta}_2\sin(\theta_2)$	$\dot{\theta}_2\cos(\theta_1)$	$\dot{\theta}_2\cos(\theta_2)$	$\dot{\theta}_2\sin^2(\theta_1)$	$\dot{\theta}_2\sin(\theta_1)\sin(\theta_2)$	$\dot{\theta}_2\sin^2(\theta_2)$
$\dot{\theta}_2\sin(\theta_1)\cos(\theta_1)$	$\dot{\theta}_2\sin(\theta_2)\cos(\theta_1)$	$\dot{\theta}_2\cos^2(\theta_1)$	$\dot{\theta}_2\sin(\theta_1)\cos(\theta_2)$	$\dot{\theta}_2\sin(\theta_2)\cos(\theta_2)$	$\dot{\theta}_2\cos(\theta_1)\cos(\theta_2)$	$\dot{\theta}_2\cos^2(\theta_2)$	$\dot{\theta}_2^2\sin(\theta_1)$	$\dot{\theta}_2^2\sin(\theta_2)$	$\dot{\theta}_2^2\cos(\theta_1)$
$\dot{\theta}_1^2\cos(\theta_2)$	$\dot{\theta}_1^2\sin^2(\theta_1)$	$\dot{\theta}_1^2\sin(\theta_1)\sin(\theta_2)$	$\dot{\theta}_1^2\sin^2(\theta_2)$	$\dot{\theta}_1^2\sin(\theta_1)\cos(\theta_1)$	$\dot{\theta}_1^2\sin(\theta_2)\cos(\theta_1)$	$\dot{\theta}_1^2\cos^2(\theta_1)$	$\dot{\theta}_1^2\sin(\theta_1)\cos(\theta_2)$	$\dot{\theta}_1^2\sin(\theta_2)\cos(\theta_2)$	$\dot{\theta}_1^2\cos(\theta_1)\cos(\theta_2)$
$\dot{\theta}_1^2\cos^2(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_1)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\cos(\theta_1)$	$\dot{\theta}_1\dot{\theta}_2\cos(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\sin^2(\theta_1)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_1)\sin(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\sin^2(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_1)\cos(\theta_1)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_2)\cos(\theta_1)$
$\dot{\theta}_1\dot{\theta}_2\cos^2(\theta_1)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_1)\cos(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\sin(\theta_2)\cos(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\cos(\theta_1)\cos(\theta_2)$	$\dot{\theta}_1\dot{\theta}_2\cos^2(\theta_2)$	$\dot{\theta}_2^2\sin(\theta_1)$	$\dot{\theta}_2^2\sin(\theta_2)$	$\dot{\theta}_2^2\cos(\theta_1)$	$\dot{\theta}_2^2\cos(\theta_2)$	$\dot{\theta}_2^2\sin^2(\theta_1)$
$\dot{\theta}_2^2\sin(\theta_1)\sin(\theta_2)$	$\dot{\theta}_2^2\sin^2(\theta_2)$	$\dot{\theta}_2^2\sin(\theta_1)\cos(\theta_1)$	$\dot{\theta}_2^2\sin(\theta_2)\cos(\theta_1)$	$\dot{\theta}_2^2\cos^2(\theta_1)$	$\dot{\theta}_2^2\sin(\theta_1)\cos(\theta_2)$	$\dot{\theta}_2^2\sin(\theta_2)\cos(\theta_2)$	$\dot{\theta}_2^2\cos(\theta_1)\cos(\theta_2)$	$\dot{\theta}_2^2\cos^2(\theta_2)$	

(a)

$\dot{\theta}_1$	$\frac{1}{2}\dot{\theta}_1^2$	$\frac{1}{3}\dot{\theta}_1^3$	$\dot{\theta}_2$	$\frac{1}{2}\dot{\theta}_2^2$	$\frac{1}{3}\dot{\theta}_2^3$
------------------	-------------------------------	-------------------------------	------------------	-------------------------------	-------------------------------

(b)

Fig. 12 Candidate library visualization of the double pendulum: (a) the Lagrangian library and (b) the Rayleigh dissipation function library

ϕ	$\dot{\theta}$	$\dot{\phi}$	ϕ^2	$\phi\dot{\theta}$	$\dot{\theta}^2$	$\phi\dot{\phi}$	$\dot{\theta}\dot{\phi}$	$\dot{\phi}^2$	$\sin(\theta)$
$\cos(\theta)$	$\sin^2(\theta)$	$\sin(\theta)\cos(\theta)$	$\cos^2(\theta)$	$\phi\sin(\theta)$	$\phi\cos(\theta)$	$\phi\sin^2(\theta)$	$\phi\sin(\theta)\cos(\theta)$	$\phi\cos^2(\theta)$	$\dot{\theta}\sin(\theta)$
$\dot{\theta}\cos(\theta)$	$\dot{\theta}\sin^2(\theta)$	$\dot{\theta}\sin(\theta)\cos(\theta)$	$\dot{\theta}\cos^2(\theta)$	$\dot{\phi}\sin(\theta)$	$\dot{\phi}\cos(\theta)$	$\dot{\phi}\sin^2(\theta)$	$\dot{\phi}\sin(\theta)\cos(\theta)$	$\dot{\phi}\cos^2(\theta)$	$\phi^2\sin(\theta)$
$\phi^2\cos(\theta)$	$\phi^2\sin^2(\theta)$	$\phi^2\sin(\theta)\cos(\theta)$	$\phi^2\cos^2(\theta)$	$\phi\dot{\theta}\sin(\theta)$	$\phi\dot{\theta}\cos(\theta)$	$\phi\dot{\theta}\sin^2(\theta)$	$\phi\dot{\theta}\sin(\theta)\cos(\theta)$	$\phi\dot{\theta}\cos^2(\theta)$	$\dot{\theta}^2\sin(\theta)$
$\dot{\theta}^2\cos(\theta)$	$\dot{\theta}^2\sin^2(\theta)$	$\dot{\theta}^2\sin(\theta)\cos(\theta)$	$\dot{\theta}^2\cos^2(\theta)$	$\phi\dot{\phi}\sin(\theta)$	$\phi\dot{\phi}\cos(\theta)$	$\phi\dot{\phi}\sin^2(\theta)$	$\phi\dot{\phi}\sin(\theta)\cos(\theta)$	$\phi\dot{\phi}\cos^2(\theta)$	$\dot{\theta}\dot{\phi}\sin(\theta)$
$\dot{\theta}\dot{\phi}\cos(\theta)$	$\dot{\theta}\dot{\phi}\sin^2(\theta)$	$\dot{\theta}\dot{\phi}\sin(\theta)\cos(\theta)$	$\dot{\theta}\dot{\phi}\cos^2(\theta)$	$\dot{\phi}^2\sin(\theta)$	$\dot{\phi}^2\cos(\theta)$	$\dot{\phi}^2\sin^2(\theta)$	$\dot{\phi}^2\sin(\theta)\cos(\theta)$	$\dot{\phi}^2\cos^2(\theta)$	

(a)

$\dot{\phi}$	$\frac{1}{2}\dot{\phi}^2$	$\frac{1}{3}\dot{\phi}^3$	$\dot{\theta}$	$\frac{1}{2}\dot{\theta}^2$	$\frac{1}{3}\dot{\theta}^3$
--------------	---------------------------	---------------------------	----------------	-----------------------------	-----------------------------

(b)

Fig. 13 Candidate library visualization of the spherical pendulum: (a) the Lagrangian library and (b) the Rayleigh dissipation function library

References

[1] Schmid, P., 2010, "Dynamic Mode Decomposition of Numerical and Experimental Data," *J. Fluid Mech.*, **656**(7), pp. 5–28.
 [2] Benner, P., Gugercin, S., and Willcox, K., 2015, "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Rev.*, **57**(4), pp. 483–531.
 [3] Peherstorfer, B., and Willcox, K., 2016, "Data-Driven Operator Inference for Nonintrusive Projection-Based Model Reduction," *Comput. Methods Appl. Mech. Eng.*, **306**, pp. 196–215.
 [4] Kutz, J., Brunton, S., Brunton, B., and Proctor, J., 2016, *Dynamic Mode Decomposition*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
 [5] Bongard, J., and Lipson, H., 2007, "Automated Reverse Engineering of Nonlinear Dynamical Systems," *Proc. Natl. Acad. Sci.*, **104**(24), pp. 9943–9948.
 [6] Schmidt, M., and Lipson, H., 2009, "Distilling Free-Form Natural Laws From Experimental Data," *Science*, **324**(5923), pp. 81–85.
 [7] Brunton, S., Proctor, J., and Kutz, J., 2016, "Discovering Governing Equations From Data by Sparse Identification of Nonlinear Dynamical Systems," *Proc. Natl. Acad. Sci.*, **113**(15), pp. 3932–3937.

- [8] Rudy, S., Brunton, S., Proctor, J., and Kutz, J., 2017, "Data-Driven Discovery of Partial Differential Equations," *Sci. Adv.*, **3**(4), p. e1602614.
- [9] Mallat, R., Bonnet, V., Huo, W., Karasinski, P., Amirat, Y., Khalil, M., and Mohammed, S., 2018, "Human-Exoskeleton System Dynamics Identification Using Affordable Sensors," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, May 21–25, pp. 6759–6765.
- [10] Bruder, D., Remy, C., and Vasudevan, R., 2019, "Nonlinear System Identification of Soft Robot Dynamics Using Koopman Operator Theory," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May 21–24, pp. 6244–6250.
- [11] Fontanelli, G., Ficuciello, F., Villani, L., and Siciliano, B., 2017, "Modelling and Identification of the da Vinci Research Kit Robotic Arms," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, Sept. 24–28, pp. 1464–1469.
- [12] Lusch, B., Kutz, J., and Brunton, S., 2018, "Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics," *Nat. Commun.*, **9**(1), p. 4950.
- [13] Wehmeyer, C., and Noé, F., 2018, "Time-Lagged Autoencoders: Deep Learning of Slow Collective Variables for Molecular Kinetics," *J. Chem. Phys.*, **148**(24), p. 241703.
- [14] Yilmaz, N., Wu, J., Kazanzides, P., and Tumerdem, U., 2020, "Neural Network Based Inverse Dynamics Identification and External Force Estimation on the da Vinci Research Kit," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, May 31–Aug. 31, pp. 1387–1393.
- [15] Sanyal, S., and Roy, K., 2023, "RAMP-Net: A Robust Adaptive MPC for Quadrotors Via Physics-Informed Neural Network," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, May 29–June 2, pp. 1019–1025.
- [16] Fasel, U., Kaiser, E., Kutz, J., Brunton, B., and Brunton, S., 2021, "SINDy With Control: A Tutorial," 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, Dec. 14–17, pp. 16–21.
- [17] Dam, M., Brøns, M., Juul Rasmussen, J., Naulin, V., and Hesthaven, J., 2017, "Sparse Identification of a Predator-Prey System From Simulation Data of a Convection Model," *Phys. Plasmas*, **24**(2), p. 022310.
- [18] De Silva, B., Higdon, D., Brunton, S., and Kutz, J., 2020, "Discovery of Physics From Data: Universal Laws and Discrepancies," *Front. Artif. Intell.*, **3**, p. 25.
- [19] Hoffmann, M., Fröhner, C., and Noé, F., 2019, "Reactive SINDy: Discovering Governing Reactions From Concentration Data," *J. Chem. Phys.*, **150**(2), p. 025101.
- [20] Chen, J., Zhang, M., Yang, Z., and Xia, L., 2021, "A Robust Data-Driven Approach for Dynamics Model Identification in Trajectory Planning," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, Sept. 27–Oct. 1, pp. 7104–7111.
- [21] Sorokina, M., Sygletos, S., and Turitsyn, S., 2016, "Sparse Identification for Nonlinear Optical Communication Systems: SINO Method," *Opt. Express*, **24**(26), p. 30433.
- [22] Cai, Y., Wang, X., Joós, G., and Kamwa, I., 2023, "An Online Data-Driven Method to Locate Forced Oscillation Sources From Power Plants Based on Sparse Identification of Nonlinear Dynamics (SINDy)," *IEEE Trans. Power Syst.*, **38**(3), pp. 2085–2099.
- [23] Huang, S.-C., Chen, H.-W., and Tien, M.-H., 2023, "Predicting Nonlinear Modal Properties by Measuring Free Vibration Responses," *ASME J. Comput. Nonlinear Dyn.*, **18**(4), p. 041005.
- [24] Kanki, R., and Saito, A., 2024, "Data-Driven Initial Gap Identification of Piecewise-Linear Systems Using Sparse Regression and Universal Approximation Theorem," *ASME J. Comput. Nonlinear Dyn.*, **19**(6), p. 061003.
- [25] Kaheman, K., Kutz, J., and Brunton, S., 2020, "SINDy-PI: A Robust Algorithm for Parallel Implicit Sparse Identification of Nonlinear Dynamics," *Proc. Math. Phys. Eng. Sci.*, **476**(2242), p. 20200279.
- [26] Chu, H., and Hayashibe, M., 2020, "Discovering Interpretable Dynamics by Sparsity Promotion on Energy and the Lagrangian," *IEEE Rob. Autom. Lett.*, **5**(2), pp. 2154–2160.
- [27] Purnomo, A., and Hayashibe, M., 2023, "Sparse Identification of Lagrangian for Nonlinear Dynamical Systems Via Proximal Gradient Method," *Sci. Rep.*, **13**(1), p. 7919.
- [28] Nesterov, Y., 2013, "Gradient Methods for Minimizing Composite Functions," *Math. Program.*, **140**(1), pp. 125–161.
- [29] Latafat, P., Themelis, A., Stella, L., and Patrinos, P., 2024, "Adaptive Proximal Algorithms for Convex Optimization Under Local Lipschitz Continuity of the Gradient," *Math. Program.*, pp. 1–39.
- [30] Li, H., and Lin, Z., 2015, "Accelerated Proximal Gradient Methods for Nonconvex Programming," *Advances in Neural Information Processing Systems*, Vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Montreal, QC, Canada.
- [31] Beck, A., and Teboulle, M., 2009, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, **2**(1), pp. 183–202.
- [32] Sun, T., and Zhang, C.-H., 2012, "Scaled Sparse Linear Regression," *Biometrika*, **99**(4), pp. 879–898.
- [33] Fasel, U., Kutz, J., Brunton, B., and Brunton, S., 2022, "Ensemble-SINDy: Robust Sparse Model Discovery in the Low-Data, High-Noise Limit, With Active Learning and Control," *Proc. R. Soc. A: Math., Phys. Eng. Sci.*, **478**(2260), p. 20210904.
- [34] Wentz, J., and Doostan, A., 2023, "Derivative-Based SINDy (DSINDy): Addressing the Challenge of Discovering Governing Equations From Noisy Data," *Comput. Methods Appl. Mech. Eng.*, **413**, p. 116096.