Sparse Identification of Lagrangian for Nonlinear Dynamical Systems via Proximal Gradient Method

Adam Purnomo and Mitsuhiro Hayashibe

Abstract— **Distilling physical laws autonomously from data has been of great interest in many scientific areas. The sparse identification of nonlinear dynamics (SINDy) and its variations have been developed to extract the underlying governing equations from observation data. The principle of the least action governs many mechanical systems, mathematically expressed in the Lagrangian formula. Compared to the actual equation of motions, the Lagrangian is much more concise, especially for complex systems. Only a few methods have been proposed to extract the Lagrangian from measurement data so far. One of such methods, Lagrangian-SINDy, can extract the true form of Lagrangian of dynamical systems from data but suffers when noises are present. In this work, we develop an extended version of Lagrangian-SINDy (xL-SINDy) to obtain the Lagrangian of dynamical systems from noisy measurement data. We incorporate the concept of SINDy and utilize the proximal gradient method to obtain sparse expressions of the Lagrangian. We demonstrated the effectiveness of xL-SINDy against different noise levels with four nonlinear dynamics: a single pendulum, a cart-pendulum, a double pendulum, and a spherical pendulum. Furthermore, we also verified the performance of xL-SINDy against SINDy-PI, a recent robust variant of SINDy to identify implicit dynamics and rational nonlinearities. Our experiment results show that xL-SINDy is 8-20 times more robust than SINDy-PI in the presence of noise.**

I. Introduction and Related Works

Since the early modern history of humanity, scientists have always been trying to come up with models that can capture real-world phenomena. Such models are desired because they can be used to devise solutions to real-world problems. For centuries, the process of refining hypothesis and models from observation data have been conducted manually. Automating this process has long been of great interest in the scientific community.

Many attempts have been made to extract physical laws autonomously from data. With the abundance of data and cheaper yet powerful hardware, the deep learning-based methods have gained a lot of attraction and have been widely used to model and control dynamical systems [1], [2]. It has also been shown that deep learning is also capable of approximating invariant quantities from dynamical systems such as the Hamiltonian [3] and the Lagrangian [4]. However, deep learning models act as black-boxes; it does not provide insights on how each observation variable affects and relates to each other.

Recent trends, however, favor parsimonious models, models with the lowest complexity to describe the observation data. A ground-breaking work done by Schmidt and Lipson [5] shows us that it is possible to extract the governing mathematical expressions from observation data. Symbolic regression is used to find the nonlinear differential equations that describe the behavior of the system, but symbolic regression tends to be expensive. The sparse identification of nonlinear dynamics (SINDy) [6] models nonlinear differential equations of dynamics as a linear combination of nonlinear candidate functions and obtain parsimonious model through sparse regression.

While there are many applications of SINDy across different fields [7]–[10], SINDy faces certain difficulties when the dynamics contain rational functions. Including rational functions into the library of candidate functions would tremendously increase the size of the library, making the sparse regression challenging. A modification of SINDy, implicit-SINDy [11], reformulates the SINDy problem into implicit form to address this challenge, albeit this method is sensitive to noise. SINDy-PI [12] is proposed to improve the performance of implicit-SINDy in terms of noise robustness.

The principle of least action is fundamental to many dynamical systems [13]. The principle states the trajectory chosen by the system is the one that minimizes a certain cost function. This cost function is the so-called 'action,' or often known as the Lagrangian. Compared to the underlying differential equations, Lagrangian has a desirable property in which it is a single scalar quantity that contains all information to predict the behavior of the systems. In robotics, the derivation of the dynamics often starts from the Lagrangian of the systems.

Several works have been proposed to approximate the Lagrangian from data with polynomial basis functions [14], [15]. However, approximating Lagrangian with polynomial basis functions will only be useful for a particular trajectory of the system and will not likely generalize well across different initial conditions. Lagrangian-SINDy [16] is a SINDy-based method designed to extract the Lagrangian of nonlinear dynamics and is shown to be able to retrieve the true form of Lagrangian of several dynamical systems. However, the author mentioned in the paper that Lagrangian-SINDy is sensitive to noise and cannot recover the Lagrangian when the training data is corrupted even with small amounts of noise. Noise will always present in real-world systems, and developing a robust method against noise is important for real-world application.

In this work, we propose a method called extended

The authors are with the Neuro-Robotics Lab, Department of Robotics, Tohoku University, 980-8579, Sendai, Japan (email : adam.syammas.zaki.purnomo.p8@dc.tohoku.ac.jp; hayashibe@tohoku.ac.jp)

Lagrangian-SINDy (xL-SINDy) that can discover the true form of Lagrangian and is more robust in the presence of noise compared to Lagrangian-SINDy. We demonstrated the effectiveness of xL-SINDy against different noise levels in physical simulation with four dynamical systems: A single pendulum, a cart-pendulum, a double pendulum, and a spherical pendulum. This paper is organized as follows. Section II describes how we use basic ideas from previous works to formulate the problem and develop the learning method. Section III presents the results of simulation experiments on the aforementioned dynamical systems. Section IV provides closing and remarks.

II. Method

A. Problem Formulation

Inspired by the concept of SINDy [6], we consider a Lagrangian expression in a structure of a linear combination of nonlinear candidate functions. Let $\boldsymbol{q} = (q_1, q_2, ..., q_n)$ be the configuration of a system in a generalized coordinate of a system, the Lagrangian of the system is expressed as

$$\mathcal{L} = \sum_{k=1}^{p} c_k \phi_k(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{1}$$

where, $\phi_k(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $k = 1, ..., p$ are a set of nonlinear candidate functions, and c_k, $k = 1, ..., p$ are the corresponding coefficients. We are interested to find the value of $\boldsymbol{c} = (c_1, c_2, ..., c_p)$ where we believe that the majority of the coefficients are zero. The Lagrangian of the system satisfies the Euler-Lagrange equations given by

$$\frac{d}{dt}\nabla_{\dot{\boldsymbol{q}}}\mathcal{L} - \nabla_{\boldsymbol{q}}\mathcal{L} = \boldsymbol{\tau}_{ext}, \tag{2}$$

where $(\nabla_{\boldsymbol{q}})_i \equiv \frac{\partial}{\partial q_i}$. We consider three different scenarios:

- Case I : External input $\boldsymbol{\tau}_{ext}$ of the system is provided.
- Case II : No external input is provided.
- Case III : Prior Lagrangian knowledge of a simpler system that forms a constituent of the system is provided.

1) With External Input: In the case where input $\boldsymbol{\tau}_{ext}$ is provided, substituting (1) in (2) yields

$$\frac{d}{dt}\sum_{k=1}^{p} c_k \nabla_{\dot{\boldsymbol{q}}}\phi_k - \sum_{k=1}^{p} c_k \nabla_{\boldsymbol{q}}\phi_k = \boldsymbol{\tau}_{pred}, \tag{3}$$

where $\boldsymbol{\tau}_{pred}$ is the predicted value of the external input $\boldsymbol{\tau}_{ext}$ given a set of coefficient $\boldsymbol{c} = (c_1, c_2, ..., c_p)$. We can further expand the time derivative $\frac{d}{dt}$ by using chain rule, giving us the terms $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$

$$\begin{aligned} \left(\sum_{k=1}^{p} c_k \nabla_{\dot{\boldsymbol{q}}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_k\right)\ddot{\boldsymbol{q}} + \left(\sum_{k=1}^{p} c_k \nabla_{\boldsymbol{q}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_k\right)\dot{\boldsymbol{q}} - \\ \left(\sum_{k=1}^{p} c_k \nabla_{\boldsymbol{q}}\phi_k\right) = \boldsymbol{\tau}_{pred}. \end{aligned} \tag{4}$$

To avoid verbose notation, we define the following notations

$$\mathbf{M}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{p} c_k \nabla_{\dot{\boldsymbol{q}}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_k, \tag{5}$$

$$\mathbf{N}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{p} c_k \nabla_{\boldsymbol{q}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_k, \tag{6}$$

$$\mathbf{O}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{p} c_k \nabla_{\boldsymbol{q}}\phi_k. \tag{7}$$

Substituting (5), (6), and (7) in (4) yields

$$\boldsymbol{\tau}_{pred} = \mathbf{M}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \mathbf{N}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \mathbf{O}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{8}$$

and we define the following cost function that we want to minimize to obtain the Lagrangian of the system

$$J(\boldsymbol{c}) = \|\boldsymbol{\tau}_{ext} - \boldsymbol{\tau}_{pred}(\boldsymbol{c})\|_2^2. \tag{9}$$

2) Without External Input: In the case of passive systems, in which no external input $\boldsymbol{\tau}_{ext}$ is not provided, eq. (8) can be modified so that we can solve for $\ddot{\boldsymbol{q}}_{pred}$ expressed as

$$\ddot{\boldsymbol{q}}_{pred} = \mathbf{M}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})^{-1}\left(\mathbf{O}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathbf{N}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right), \tag{10}$$

where $\ddot{\boldsymbol{q}}_{pred}$ represents the predicted value of acceleration $\ddot{\boldsymbol{q}}$ and $(\cdot)^{-1}$ represents matrix inverse. In practice, we use Moore-Penrose pseudo inverse to calculate eq. (10) since there is no guarantee that the matrix is not a singular matrix. We define the following cost function to learn the Lagrangian of the system

$$J(\boldsymbol{c}) = \|\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{pred}(\boldsymbol{c})\|_2^2. \tag{11}$$

Solving the optimization for the above cost function is really challenging due to the inverse operation, hence making equation (11) is highly non-convex with respect to variable $\boldsymbol{c}$. We empirically found that with a library of more than 20 candidate functions, the learning process will hardly converge even after a long period of iterations. Therefore, this problem usually only works for simple systems such as a single pendulum. For more complex systems, it is preferable that either the external input $\boldsymbol{\tau}_{ext}$ is provided, or prior Lagrangian knowledge of a simpler system that forms a constituent of the larger system is provided to boost the learning process which will be explained in the next paragraph.

3) With Prior Knowledge: For multi-DOF non-relativistic systems, the Lagrangian can be described as $\mathcal{L} = \sum_i T_i - \sum_i V_i = \sum_i (T_i - V_i)$, where T_i and V_i are the kinetic energy and potential energy of each constituent of the system. Since the total Lagrangian of the system is the sum of Lagrangian of its constituents, it is reasonable to assume that the nonlinear terms that appear in each constituent will also appear in the total Lagrangian of the system [16].

Given prior knowledge of a constituent of the system, we pick one out of several terms that appear in the total Lagrangian of the system and label them as $\phi_r(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

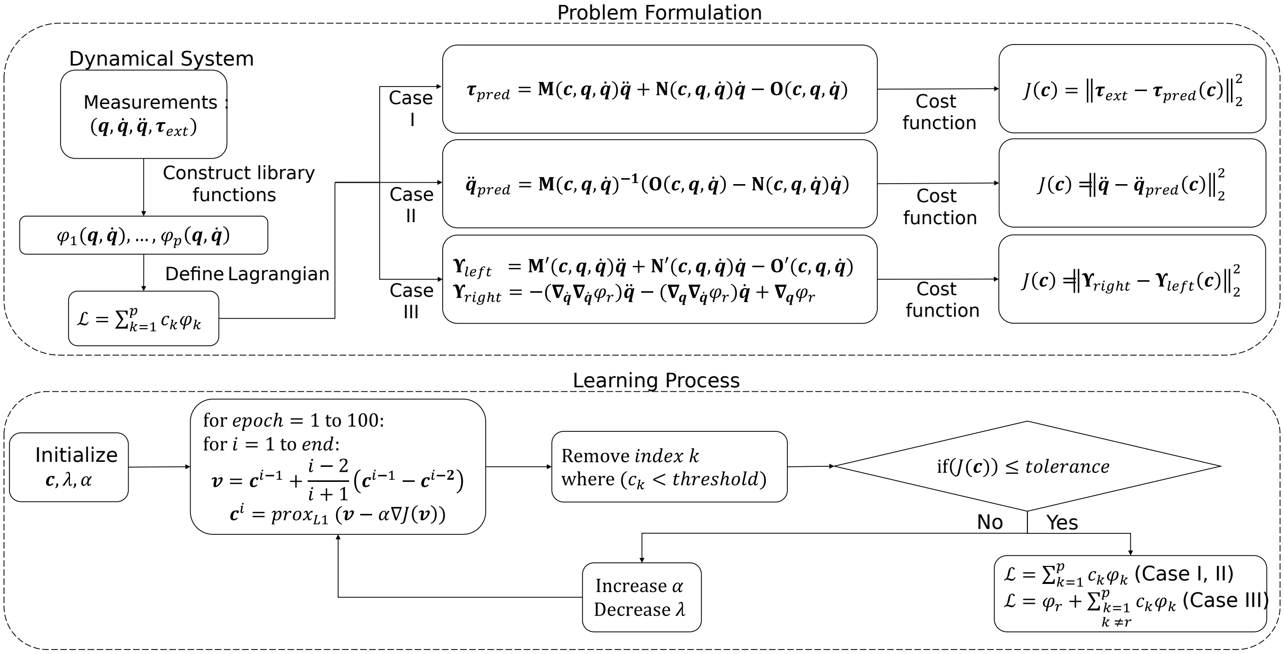
Problem Formulation
Dynamical System
Measurements : $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\tau}_{ext})$
Construct library functions
$\varphi_1(\boldsymbol{q}, \dot{\boldsymbol{q}}), \dots, \varphi_p(\boldsymbol{q}, \dot{\boldsymbol{q}})$
Define Lagrangian
$\mathcal{L} = \sum_{k=1}^{p} c_k \varphi_k$
Case I
Case II
Case III
$\boldsymbol{\tau}_{pred} = \mathbf{M}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \mathbf{N}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \mathbf{O}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})$
$\ddot{\boldsymbol{q}}_{pred} = \mathbf{M}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})^{-1}(\mathbf{O}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathbf{N}(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}})$
$\boldsymbol{\Upsilon}_{left} = \mathbf{M}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \mathbf{N}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \mathbf{O}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})$
$\boldsymbol{\Upsilon}_{right} = -(\boldsymbol{\nabla}_{\dot{\boldsymbol{q}}}\boldsymbol{\nabla}_{\dot{\boldsymbol{q}}}\varphi_r)\ddot{\boldsymbol{q}} - (\boldsymbol{\nabla}_{\boldsymbol{q}}\boldsymbol{\nabla}_{\dot{\boldsymbol{q}}}\varphi_r)\dot{\boldsymbol{q}} + \boldsymbol{\nabla}_{\boldsymbol{q}}\varphi_r$
Cost function
Cost function
Cost function
$J(\boldsymbol{c}) = \|\boldsymbol{\tau}_{ext} - \boldsymbol{\tau}_{pred}(\boldsymbol{c})\|_2^2$
$J(\boldsymbol{c}) = \|\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{pred}(\boldsymbol{c})\|_2^2$
$J(\boldsymbol{c}) = \|\boldsymbol{\Upsilon}_{right} - \boldsymbol{\Upsilon}_{left}(\boldsymbol{c})\|_2^2$
Learning Process
Initialize $\boldsymbol{c}, \lambda, \alpha$
for $epoch = 1$ to 100:
for $i = 1$ to end:
$\boldsymbol{v} = \boldsymbol{c}^{i-1} + \frac{i-2}{i+1}(\boldsymbol{c}^{i-1} - \boldsymbol{c}^{i-2})$
$\boldsymbol{c}^i = prox_{L1}\,(\boldsymbol{v} - \alpha\nabla J(\boldsymbol{v}))$
Remove $index\ k$ where $(c_k < threshold)$
if$(J(\boldsymbol{c})) \leq tolerance$
No
Yes
Increase α
Decrease λ
$\mathcal{L} = \sum_{k=1}^{p} c_k \varphi_k$ (Case I, II)
$\mathcal{L} = \varphi_r + \sum_{k=1, k \neq r}^{p} c_k \varphi_k$ (Case III)

Fig. 1: Block diagram of the proposed method (xL-SINDy). Depending on the case of the problem, a different cost function is constructed. Once the cost function is defined, the cost function is minimized by using proximal gradient descent method.

The Lagrangian of a system is not unique; many forms of Lagrangians can satisfy Euler-Lagrange's equation for a particular system. For example, $\mathcal{L}' = k\mathcal{L}$, where k is a constant, still satisfies the Euler-Lagrange's equation. By multiplying equation (1) with $k = \frac{1}{c_r}$, eq. (1) can be modified as

$$\mathcal{L} = \phi_r(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \sum_{\substack{k=1 \\ k \neq r}}^{p} c_k' \phi_k(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{12}$$

where $c_k' = \frac{c_k}{c_r}$. Now, the variable c_k' becomes the coefficients that we are interested to solve. We can simply redefine $c_k := c_k'$ for notation simplicity. The Euler-Lagrange equation of the system can be expressed as

$$\frac{d}{dt}\sum_{\substack{k=1 \\ k \neq r}}^{p} c_k \nabla_{\dot{\boldsymbol{q}}}\phi_k - \sum_{\substack{k=1 \\ k \neq r}}^{p} c_k \nabla_{\boldsymbol{q}}\phi_k = -\frac{d}{dt}\nabla_{\dot{\boldsymbol{q}}}\phi_r + \nabla_{\boldsymbol{q}}\phi_r. \tag{13}$$

We define the following notation

$$\begin{aligned} \boldsymbol{\Upsilon}_{left} &= \frac{d}{dt}\sum_{\substack{k=1 \\ k \neq r}}^{p} c_k \nabla_{\dot{\boldsymbol{q}}}\phi_k - \sum_{\substack{k=1 \\ k \neq r}}^{p} c_k \nabla_{\boldsymbol{q}}\phi_k \\ &= \mathbf{M}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \mathbf{N}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \mathbf{O}'(\boldsymbol{c}, \boldsymbol{q}, \dot{\boldsymbol{q}}), \end{aligned} \tag{14}$$

$$\begin{aligned} \boldsymbol{\Upsilon}_{right} &= -\frac{d}{dt}\nabla_{\dot{\boldsymbol{q}}}\phi_r + \nabla_{\boldsymbol{q}}\phi_r \\ &= -\left(\nabla_{\dot{\boldsymbol{q}}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_r\right)\ddot{\boldsymbol{q}} - \left(\nabla_{\boldsymbol{q}}^{\top}\nabla_{\dot{\boldsymbol{q}}}\phi_r\right)\dot{\boldsymbol{q}} + \nabla_{\boldsymbol{q}}\phi_r, \end{aligned} \tag{15}$$

where $\boldsymbol{\Upsilon}_{left}$ and $\boldsymbol{\Upsilon}_{right}$ represent the left hand side (LHS) and the right hand side (RHS) of eq. (13). By minimizing the following cost function

$$J(\boldsymbol{c}) = \|\boldsymbol{\Upsilon}_{right} - \boldsymbol{\Upsilon}_{left}(\boldsymbol{c})\|_2^2, \tag{16}$$

it is possible to obtain the true Lagrangian of the system. We usually have more than one option of ϕ_r to construct $\boldsymbol{\Upsilon}_{right}$. In practice, we have to test all of them one by one and choose the one that yields the best model.

B. Learning Lagrangian

The proposed learning method to obtain the Lagrangian is summarized in Fig. 1. We start with the problem formulation as described in the previous section. Given a dynamical system, we gather times series data $\{t_i, \boldsymbol{q}(t_i), \dot{\boldsymbol{q}}(t_i), \ddot{\boldsymbol{q}})(t_i, \boldsymbol{\tau}_{ext}(t_i)\}_{i=1}^{N}$ from several initial conditions. We then proceed to construct a library of candidate functions.

In general, the larger the library of the candidate functions, the more difficult the optimization problem becomes. It is especially true when several candidate functions can behave in a similar manner, such as in the trigonometric family functions. It is important to carefully construct a sufficient library but not too large so that the optimization problem is still tractable. It is also important, however, not to include trivial terms that satisfy Euler-Lagrange's equation regardless of trajectories such as $\mathcal{L} = \boldsymbol{q}^n\dot{\boldsymbol{q}}$. Depending on the case of the problem, a different cost function should be defined.

As mentioned previously, we believe that the correct solution is sparse where the majority of the coefficients are zero. Therefore, we add L1 regularization term to the cost function for sparsity constraint [17] expressed as

$$J'(\boldsymbol{c}) = J(\boldsymbol{c}) + \lambda \|\boldsymbol{c}\|_1, \tag{17}$$

where λ is the sparsity promoting parameter that we have to carefully tune. In this work, we use accelerated proximal gradient descent method [18] to minimize the composite cost function defined above. Given an initial point $\boldsymbol{c}^0$, the update step of proximal gradient descent is defined as

$$\boldsymbol{v} = \boldsymbol{c}^{i-1} + \frac{i-2}{i+1}\left(\boldsymbol{c}^{i-1} - \boldsymbol{c}^{i-2}\right), \tag{18}$$

$$\boldsymbol{c}^i = prox_{L1}\left(\boldsymbol{v} - \alpha \nabla J(\boldsymbol{v})\right), \tag{19}$$

where c^i is the coefficient $\boldsymbol{c}$ at iteration i, α is the learning rate, and $prox_{L1}(\cdot)$ is the proximal operator for L1 norm. The L1 norm penalty term is a separable sum of the component of its input, and a proximal operator is used to minimize this term. The proximal operator for L1 norm is well defined separately for each component of the input and expressed as follows,

$$[prox_{L1}(\boldsymbol{\beta}]_k = \text{sign}(\beta_k)\max(|\beta_k| - \lambda, 0), \tag{20}$$

where k is the k^{th} entry of the input vector $\boldsymbol{\beta}$. As for case III, if we know other terms that appear in the Lagrangian but are not used to construct $\boldsymbol{\Upsilon}_{right}$, we don't put a penalty on these terms by not applying the proximal operator in eq. (20) for index k corresponding with these terms.

We proceed to initialize the value of the coefficient $\boldsymbol{c}$, the learning rate α, and the L1 norm penalty parameter λ. The learning process is done in several stages, with 100 epochs and batch size equal to 128 for each stage, until the cost function reaches the defined tolerance value as shown in Fig. 1. The tolerance value is ideally at 10^{-3}. However, converging to this value might not be possible in the presence of noise, and we have to relax the tolerance value; otherwise, the algorithm will never stop. In the beginning, the number of candidate functions in the library is usually large, and we want to eliminate non-relevant candidate functions as much as possible during the first learning stage. Therefore, we initially set the value of λ to be quite high, which is between 1 and 5.

It is important to note that every candidate function may have different magnitude scales. The L1 norm penalty penalizes all terms equally regardless of the magnitude scale, resulting in candidate functions with smaller magnitude scales being penalized more. It may or may not be necessary to do scaling in the first learning stage, where the value of λ is high, by multiplying each candidate function with scaling term s_k in eq. (1) for the case I and II, or eq. (12) for case III depending on the differences of magnitude scale between each candidate function. The learning rate is also an important hyper-parameter, especially during the first learning stage. We found that a high learning rate in the initial stage can cause the relevant terms to be penalized, preventing the model from obtaining the true Lagrangian of the system. During the initial stage, we set the learning rate $\alpha <= 10^{-5}$.

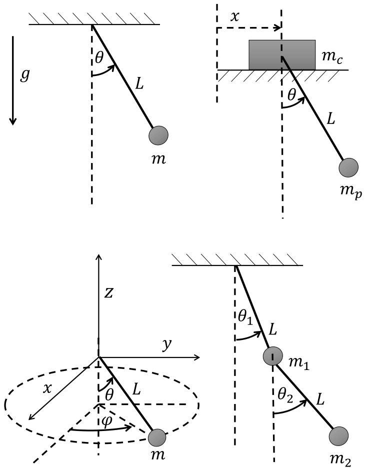

Fig. 2: Dynamical systems used to verify xL-SINDy. From upper left to bottom right: A single pendulum, a cart-pendulum, a spherical pendulum, and a double pendulum. For all systems, the length of the rod is $L = 1.0$ m, the mass of all pendulums are $m = m_p = m_1 = m_2 = 1.0$ kg, and the gravitational acceleration is $g = 9.81$ m/s^2. For the cart-pendulum, the mass of the cart is $m_c = 0.5$ kg.

At the end of every learning stage, we perform hard-thresholding by removing index k from eq. (1) or (12), where the value of $c_k < threshold$. This step effectively reduces the number of candidate functions considered in the learning process, making the convergence much faster than if hard-thresholding were not performed. We then check whether the cost function has reached the tolerance or not. If it is the latter, we proceed to the next learning stage. With fewer candidate functions after the previous hard-thresholding process, we can decrease the value of λ and increment the learning rate α to speed up the learning process. This step is repeated over and over again until the cost function reaches the tolerance value. Once the tolerance value is reached, we compute the value of the coefficient to the eq. (1) for the case I and II, or (12) for case III, and we obtain the analytical form of Lagrangian of the system.

C. Dynamical Systems and Experiments

We evaluated xL-SINDy with four ideal dynamical systems as shown in Fig. 2. In this work, we focus on the case of passive systems where no external input $\boldsymbol{\tau}_{ext}$ is provided. No prior knowledge is used for a single pendulum in the learning process, and we use the computation described by case II. For the cart-pendulum, double pendulum, and spherical pendulum, we use the computation described by case III under the assumption that we already obtained the Lagrangian of a single pendulum.

For each system, we collect training data by performing simulation with 100 initial conditions for a period of 5s each and 100 Hz of measurement frequency. After obtaining the analytical form of the Lagrangian, we create a validation data set to test the obtained model by calculating the pre-

TABLE I: Extracted Lagrangian from simulation data with various noise levels

Noise Magnitude	**Physical systems**			
	Single Pendulum	***Cart Pendulum***	***Double pendulum***	***Spherical Pendulum***
True Model	$0.500\dot{\theta}^2 + 9.810\cos\theta$	$0.250\dot{\theta}^2 + 0.750\dot{x}^2$ $+ 0.500\dot{x}\dot{\theta}\cos\theta$ $+ 4.905\cos\theta$	$19.620\cos\theta_1 + 9.810\cos\theta_2$ $+ 1.000\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 1.000\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.500\dot{\theta_2}^2$	$0.500\dot{\phi}^2\sin^2\theta + 0.500\dot{\theta}^2$ $+ 9.810\cos\theta$
$\sigma = 0$	$0.295\dot{\theta}^2 + 5.797\cos\theta$	$1.000\dot{\theta}^2 + 2.975\dot{x}^2$ $+ 1.984\dot{x}\dot{\theta}\cos\theta$ $+ 19.755\cos\theta$	$19.620\cos\theta_1 + 9.750\cos\theta_2$ $+ 1.000\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 0.999\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.499\dot{\theta_2}^2$	$1.000\dot{\phi}^2\sin^2\theta + 1.000\dot{\theta}^2$ $+ 19.630\cos\theta$
$\sigma = 10^{-3}$	$0.268\dot{\theta}^2 + 5.252\cos\theta$	$1.000\dot{\theta}^2 + 2.975\dot{x}^2$ $+ 1.984\dot{x}\dot{\theta}\cos\theta$ $+ 19.756\cos\theta$	$19.508\cos\theta_1 + 9.755\cos\theta_2$ $+ 1.000\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 0.999\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.499\dot{\theta_2}^2$	$1.000\dot{\phi}^2\sin^2\theta + 1.000\dot{\theta}^2$ $+ 19.630\cos\theta$
$\sigma = 2 \times 10^{-2}$	$0.334\dot{\theta}^2 + 6.540\cos\theta$	$1.000\dot{\theta}^2 + 2.993\dot{x}^2$ $+ 1.994\dot{x}\dot{\theta}\cos\theta$ $+ 19.534\cos\theta$	$19.545\cos\theta_1 + 9.770\cos\theta_2$ $+ 1.000\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 0.999\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.499\dot{\theta_2}^2$	$1.000\dot{\phi}^2\sin^2\theta + 1.000\dot{\theta}^2$ $+ 19.600\cos\theta$
$\sigma = 6 \times 10^{-2}$	$0.557\dot{\theta}^2 + 10.938\cos\theta$	$1.000\dot{\theta}^2 + 1.696\dot{x}^2$ $+ 1.136\dot{x}\dot{\theta}\cos\theta$ $+ 18.082\cos\theta$ $- 0.121\dot{x}^2\cos\theta$ $+ 1.463\cos^3\theta$ *	$19.541\cos\theta_1 + 9.753\cos\theta_2$ $+ 0.999\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 0.999\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.496\dot{\theta_2}^2$	$0.130\dot{\phi}^2\sin^2\theta + 1.000\dot{\theta}^2$ $+ 2.350\cos\theta$ $- 0.790\dot{\theta}^2\sin\theta$ $- 0.430\dot{\theta}^2\cos\theta$ *
$\sigma = 10^{-1}$	$0.085\dot{\theta}^2 + 1.540\cos\theta$ $- 0.129\dot{\theta}\sin\theta$ $+ 0.551\sin^2\theta$ $- 0.019\theta^2$ *	$1.000\dot{\theta}^2 + 1.562\dot{x}^2 +$ $1.050\dot{x}\dot{\theta}\cos\theta$ $+ 19.504\cos\theta$ $- 0.143\dot{\theta}^2\cos\theta$ *	$19.381\cos\theta_1 + 9.679\cos\theta_2$ $+ 0.998\dot{\theta_1}\dot{\theta_2}\cos\theta_1\cos\theta_2$ $+ 0.992\dot{\theta_1}\dot{\theta_2}\sin\theta_1\sin\theta_2$ $+ 1.000\dot{\theta_1}^2 + 0.495\dot{\theta_2}^2$	$- 0.2\dot{\phi}^2\sin^2\theta + 1.000\dot{\theta}^2$ $+ 5.12\cos\theta$ $- 1.100\dot{\theta}^2\sin\theta$ $- 0.560\dot{\theta}^2\cos\theta$ $- 0.055\dot{\phi}^2\sin 2\theta$ *

*Terms highlighted with red color are extra terms that are not supposed to be included in the Lagrangian.
All numbers are rounded to 3 decimal places.

dicted states for the accuracy evaluation. We compute the Euler-Lagrange's equation with the obtained model, retrieve the differential equation of the system, and integrate the equations to compare it with the actual validation data. We also tested our method with training data that are corrupted by zero-mean white Gaussian noise $\mathcal{N}(0, \sigma)$ on different scale magnitude in the range of $10^{-8} <= \sigma <= 10^{-1}$. Finally, we also compare the performance of xL-SINDy on cart-pendulum, double pendulum, and sphere pendulum with noisy training data against Lagrangian-SINDy [16], and SINDy-PI [12].

III. Results

In this section, we demonstrate the effectiveness of xL-SINDy with the aforementioned nonlinear dynamical systems against various noise levels. The obtained Lagrangian for each system is summarized in Table. I. The accuracy of xL-SINDy and SINDy-PI from our simulation experiments on a cart pendulum, a double pendulum, and a spherical pendulum is shown in Fig. 3. In the second column of the plot in Fig. 3, where the noise magnitude is $\sigma = 2 \times 10^{-2}$, we can observe that SINDy-PI already starts to deviate from the true models in all three dynamical systems. At the same noise magnitude, xL-SINDy still predicts accurate models. It is also good to note that the model estimate of xL-SINDy is still reasonable even though wrong additional terms are included in the Lagrangian from the example of the cart pendulum under the noise magnitude of $\sigma = 6 \times 10^{-2}$. It indicates that the model estimate is potentially usable even when an incorrect Lagrangian structure is discovered. Our simulation results demonstrate that xL-SINDy has notably better prediction accuracy compared to SINDy-PI in the presence of higher noise magnitude in all three dynamical systems.

A. Single Pendulum

The state of a single pendulum is described be $[\theta, \dot{\theta}]$, and the Lagrangian expression of a single pendulum is given by $\mathcal{L} = \frac{1}{2}m\dot{\theta}^2 + mg\cos\theta$, Substituting the parameter given

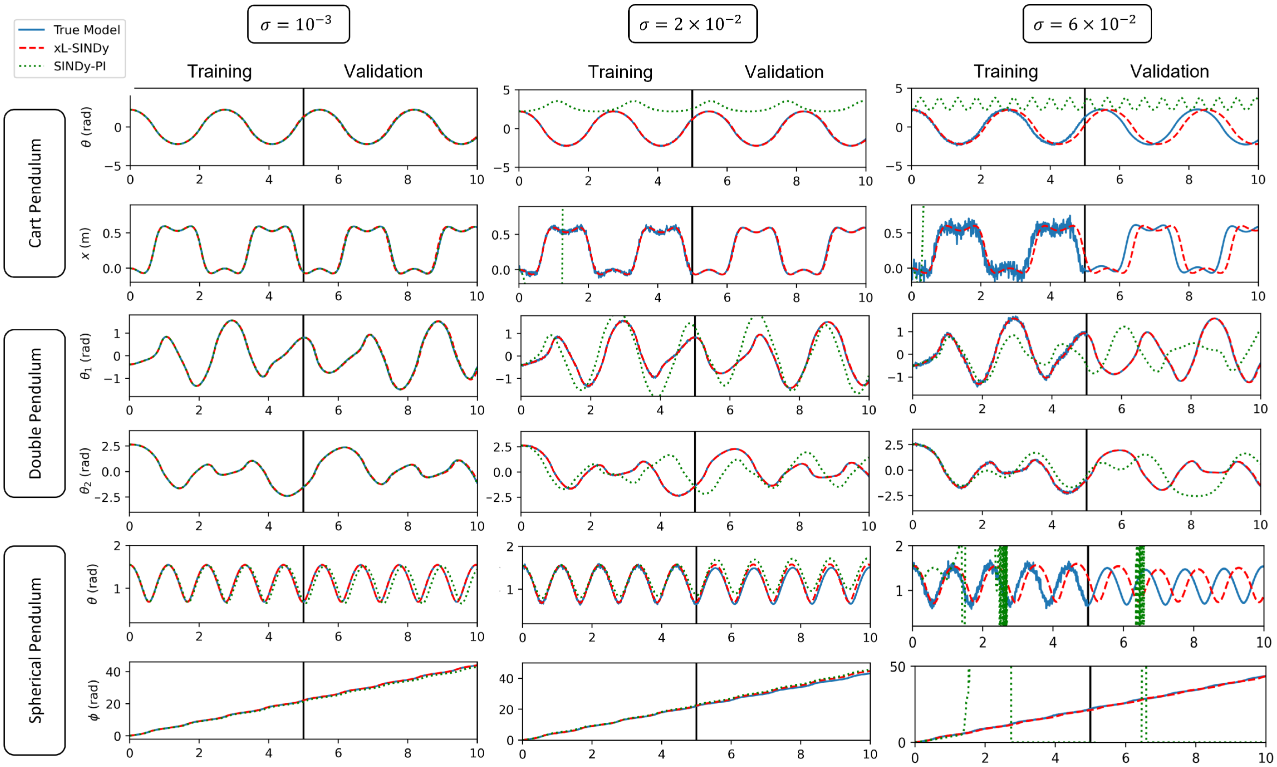

Fig. 3: Simulation results against three different noise levels for three different models: the true model, model discovered by the proposed method, and model discovered by SINDy-PI. Training data consists of 100 initial conditions in a time period of 5 seconds each. Validation (extrapolation beyond the training data set) is conducted for 5 seconds afterward. The results shown are taken randomly from one of the initial conditions from the training data set for cart pendulum, double pendulum, and spherical pendulum.

in Fig. 2, the true Lagrangian expression is shown in the second row and second column of Table. I. To construct a library of candidate functions, we create a polynomial combination of $\{\theta, \dot{\theta}, \cos\theta, \sin\theta\}$ up to the second order while excluding trivial terms such as $\dot{\theta}$ and $\theta\dot{\theta}$ resulting in 12 candidate functions. Training data with initial conditions of $[-\pi < \theta < \pi, 0]$ are created.

The initial value of the hyperparameters are $\alpha = 10^{-5}$ and $\lambda = 0.1$. The cut-off threshold is 10^{-2} for the initial learning stage and 10^{-1} for the subsequent learning stages. In the subsequent learning stages, α is increased by a factor of 2, and λ is decreased by a factor of 10. The training converged in three stages for noise magnitude $\sigma <= 10^{-3}$, and four stages for higher magnitude with a relaxed tolerance value. The correct Lagrangian structure, ones without additional terms or missing terms compared to the true Lagrangian form, can be obtained in the presence of noise magnitude up to $\sigma = 6 \times 10^{-2}$. Even though the coefficients obtained differ from the true model, the ratio of coefficients between the two terms is close compared to the true model.

B. Cart Pendulum

The state of cart pendulum is represented as $[\theta, \dot{\theta}, x, \dot{x}]$, and the Lagrangian with numerical coefficients is shown in the second row and third column of Table. I with parameters given by Fig. 2. A library of candidate function with a polynomial combination of $\{\dot{\theta}, \cos\theta, \sin\theta, x, \dot{x}\}$ up to the third order is constructed, resulting in 55 candidate functions.We here exclude the term θ because this term does not appear in the Lagrangian of a single pendulum system. Training data with initial conditions of $[-\pi < \theta < \pi, 0, 0, 0]$ are created.

The Lagrangian of a single pendulum contains $\dot{\theta}^2$ and $\cos\theta$. Hence, both terms will also appear in the Lagrangian of the cart pendulum. We tested both $\dot{\theta}^2$ and $\cos\theta$ to construct $\mathbf{\Upsilon}_{right}$ as described in eq. (15), and we found that the term $\dot{\theta}^2$ gives better results. The initial value of the hyperparameters are $\alpha = 10^{-5}$ and $\lambda = 1$. The cut-off threshold, the increment of α, and the decrement of λ are the same as the previous case. The training converged in three stages for noise magnitude $\sigma <= 2 \times 10^{-2}$, and four stages for higher magnitude with a relaxed tolerance value. xL-SINDy can recover the correct structure of the Lagrangian with noise magnitude up to $\sigma = 4 \times 10^{-2}$. In contrast, even in the case of external input force on the cart, SINDy-PI can only recover the correct structure with noise magnitude up to $\sigma = 5 \times 10^{-3}$. Therefore, in the case of the cart pendulum, xL-SINDy is 8 times more robust than SINDy-PI in the presence of noise.

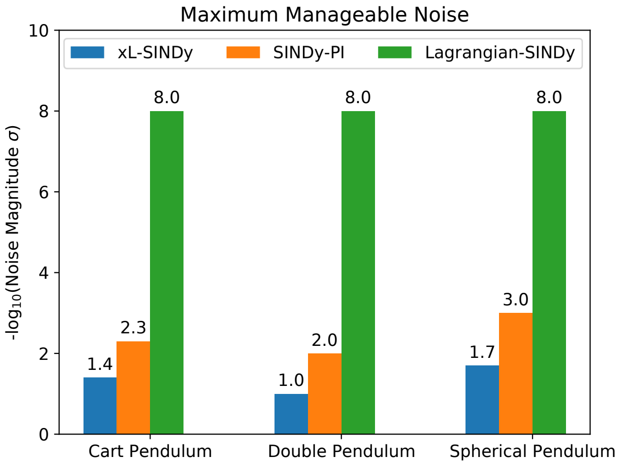

Fig. 4: Comparison of the maximum level of manageable noise in training data before an incorrect model structure is discovered for three methods: xL-SINDy, SINDy-PI, Lagrangian-SINDy. The noise level is transformed in $-\log_{10}$ scale, and a lower value in the scale represents a larger noise level. xL-SINDy provides the most robust performance against noisy training data in all simulations of three different dynamical systems.

C. Double Pendulum

Given the state of a double pendulum, $[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]$ and the system parameters in Fig. 2, the expression of the Lagrangian with numerical coefficients is shown in the second row and fourth column of Table. I. To build a library of candidate functions, we first separate the set of trigonometric terms $\{\cos\theta_1, \sin\theta_1, \sin\theta_1, \sin\theta_2\}$, and the non trigonometric terms $\{\dot{\theta}_1, \dot{\theta}_2\}$. For each set, we create a polynomial combination up to the second order, resulting in 14 candidate functions and 5 candidate functions respectively. We then generate cross terms between the two sets creating 70 candidate functions, and we have in total 89 candidate functions in the library. Training data are created with initial conditions of $[-\pi < \theta_1 < \pi, -\pi < \theta_2 < \pi, 0, 0]$.

Both constituents of double pendulum are a single pendulum. Hence, we have 4 options to construct Υ_{right}: $\dot{\theta_1}^2$, $\dot{\theta_2}^2$, $\cos\theta_1$, and $\cos\theta_2$. Both $\dot{\theta_1}^2$ and $\dot{\theta_2}^2$ yield equally good results. The results displayed in Table. I is the one with $\dot{\theta_1}^2$ used to construct Υ_{right}. The initial value of the hyperparameters are $\alpha = 5 \times 10^{-6}$ and $\lambda = 1$. The cut-off threshold, the increment of α, and the decrement of λ are the same as in previous cases. From our experiments, xL-SINDy can identify the correct structure with noise magnitude up to $\sigma = 1 \times 10^{-1}$, while SINDy-PI can extract the correct structure of equations of motions with noise magnitude up to $\sigma = 10^{-2}$. Hence, xL-SINDy is 10 times more robust against noise than SINDy-PI in this experiment.

D. Spherical Pendulum

The state of cart pendulum is represented as $[\theta, \phi, \dot{\theta}, \dot{\phi}]$, and the true Lagrangian expression is displayed in the second row and fifth column of Table. I. As in the case of double pendulum, we first separate the trigonometric terms $\{\cos\theta, \sin\theta\}$ and the non trigonometric terms $\{\dot{\theta}, \phi, \dot{\phi}\}$, create a polynomial combinations for both sets up to the second order, and add cross terms between the two sets. In total, we have 59 candidate functions in our library. The training data are created with initial conditions of $[\pi/3 < \theta < \pi/2, 0, 0, \pi]$. We deliberately choose high value of θ and $\dot{\phi}$ as the initial conditions because the equation of motion contains $\frac{1}{\sin\theta}$ which could blow up for small value of θ.

The spherical pendulum is a higher dimensional analog of a single pendulum in the first case. Therefore, we can think of the Lagrangian of a spherical pendulum as the sum of Lagrangian of the pendulum in $\hat{\theta}$ direction and $\hat{\phi}$ direction. Since we already know the Lagrangian of a single pendulum in $\hat{\theta}$ direction, we can use $\dot{\theta}^2$ and $\cos\theta$ to construct Υ_{right}. The initial value of the hyperparameters are $\alpha = 1 \times 10^{-5}$ and $\lambda = 1$. The cut-off threshold, the increment of α, and the decrement of λ are the same as in all previous cases. In this experiment, xL-SINDy shows 20 times more robustness against noise than SINDy-PI. The correct structure can be obtained with noise magnitude up to $\sigma = 2 \times 10^{-2}$ by xL-SINDy, and $\sigma = 1 \times 10^{-3}$ by SINDy-PI.

IV. Discussion

The comparison of xL-SINDy, SINDy-PI, and Lagrangian-SINDy is summarized in Fig. 4. In all three dynamical systems used as a comparison, xL-SINDy outperforms other methods in terms of noise robustness. Our experiment results demonstrate that xL-SINDy can overcome the challenge faced by Lagrangian-SINDy. xL-SINDy is capable of discovering the correct Lagrangian expression for idealized nonlinear dynamical systems in the presence of much higher noise magnitude. On top of that, xL-SINDy successfully extracts the Lagrangian in cases where Lagrangian-SINDy fails to do so, such as the non-actuated spherical pendulum [16]. The obtained coefficients may not be precisely the same as the true models, but the ratio between coefficients of each term is close to the true models.

From our experiments, while SINDy-PI is also robust against noise up to a certain magnitude, xL-SINDy is 8 to 20 times more robust against noise. SINDy-PI attempts to seek the expression of the dynamics which may contain rational functions. To do so, SINDy-PI reformulates the problem into implicit form, and it requires the library to include candidate functions of the states and the time derivative of the states of the systems. Unlike SINDy-PI, xL-SINDy only works with non-rational functions, hence only requires the library to include functions of the states, resulting in a simpler library. On top of that, SINDy-PI may have a problem when the incorrect combination of denominator terms is discovered. In rational function, when the denominator is equal to zero, its value blows up. Indeed, this is the case of SINDy-PI in some of our experiment results when SINDy-PI cannot handle the noise and discover the wrong combination of denominator terms.

One major limitation of xL-SINDy is the difficulty in designing the library. Prior knowledge of the systems is essential to decide what candidate functions we should include in the library. A large number of candidate functions in the library are more likely to be sufficient, but it makes the sparse optimization more challenging and less robust against noise [12]. Hence, balancing this trade-off is crucial for the outcome of the learning process. Unless we develop a better way to construct the library, applying xL-SINDy to more complex systems with a higher degree of freedom is still challenging as the number of candidate functions grows rapidly.

Like other learning-based methods, SINDy-PI introduces several hyperparameters during the learning process, such as the sparsity constrain λ, the learning rate α, the tolerance for the cost function, and the cut-off threshold in the hard-thresholding process. Tuning the hyperparameters is also vital for the learning outcome, especially the initial value of learning rate α and sparsity constraint λ. The process of hyperparameter tuning is currently done manually with trial and error. Algorithms specifically designed to optimize the hyperparameters such as Optuna [19] can be used to fully automate the learning process.

So far, we have not considered the presence of external non-conservative force acting on the systems. In real-world scenarios, no matter how small, non-conservative forces such as damping or friction are always present. Taking into account this external force in the model is of importance to make xL-SINDy applicable to real systems. One possible way to incorporate non-conservative force is by using the generalized Rayleigh's dissipation function [20]. Like the Lagrangian, Rayleigh's dissipation function is a single scalar quantity and can be incorporated into Euler-Lagrange's equation. We can model the generalized Rayleigh's dissipation function as a linear combination of candidate functions and learn both Lagrangian and Rayleigh's dissipation function simultaneously.

V. Conclusion

This work introduced xL-SINDy, a method to extract the Lagrangian of nonlinear dynamical systems from noisy measurement data. We model the Lagrangian as a linear combination of nonlinear candidate functions and use Euler-Lagrange's equation to formulate the objective cost function. We use the proximal gradient method to optimize the cost function and obtain sparse expression of Lagrangian. We demonstrated the effectiveness of xL-SINDy and showed that xL-SINDy is more robust against noise compared to other methods. It is worth noting that our proposed method outperforms SINDy-PI (parallel, implicit), a recent robust variant of SINDy developed for implicit dynamics and rational nonlinearities. We believe that xL-SINDy is a promising approach for the identification of interpretable models of nonlinear dynamics. The focus of our next work is to consider non-conservative forces in the model and apply xL-SINDy to real systems.

References

[1] P. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems*, 2016, pp. 4509–4517.

[2] I. Lenz, R. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control," *Robotics: Science and Systems XI*, 2015.

[3] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Advances in Neural Information Processing Systems*, vol. 32. Neural information processing systems foundation, 2019.

[4] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," 2020.

[5] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, p. 81–85, 2009.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016. [Online]. Available: https://www.pnas.org/content/113/15/3932

[7] M. Sorokina, S. Sygletos, and S. Turitsyn, "Sparse identification for nonlinear optical communication systems: Sino method," *Optics Express*, vol. 24, 2016.

[8] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven, "Sparse identification of a predator-prey system from simulation data of a convection model," *Physics of Plasmas*, vol. 24, 2017.

[9] J. C. Loiseau and S. L. Brunton, "Constrained sparse galerkin regression," *Journal of Fluid Mechanics*, vol. 838, 2018.

[10] B. M. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz, "Discovery of physics from data: Universal laws and discrepancies," *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[11] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 2, 2016.

[12] K. Kaheman, J. N. Kutz, and S. L. Brunton, "Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, no. 2242, p. 20200279, 2020. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2020.0279

[13] Leonhard, "Methodus inveniendi lineas curvas maximi minive proprietate gaudentes." *Bousquet, Lausanne & Geneva*, 1744.

[14] D. J. Hills, A. M. Grütter, and J. J. Hudson, "An algorithm for discovering lagrangians automatically from data," *PeerJ Computer Science*, vol. 1, 2015.

[15] M. Ahmadi, U. Topcu, and C. Rowley, "Control-oriented learning of lagrangian and hamiltonian systems," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 520–525.

[16] K. Chu and M. Hayashibe, "Discovering interpretable dynamics by sparsity promotion on energy and the lagrangian," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2020.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[18] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, p. 183–202, Mar. 2009. [Online]. Available: https://doi.org/10.1137/080716542

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[20] E. Minguzzi, "Rayleigh's dissipation function at work," *European Journal of Physics*, vol. 36, no. 3, p. 035014, mar 2015. [Online]. Available: https://doi.org/10.1088/0143-0807/36/3/035014