Symbolic Regression for Generic Systems

Vivant Rodolphe
UFR de Mathématiques, University of Strasbourg
Stage M1 CSMI, 2024-2025

August 21, 2025

Abstract

Cet article présente une étude sur la régression symbolique appliquée à l'identification de systèmes génériques. Le contexte porte sur la modélisation de systèmes dynamiques à partir de données observées, avec pour problématique la découverte de modèles interprétables et parcimonieux. Nous comparons la méthode SINDy et son extension ADAM-SINDy, qui optimise simultanément les paramètres non linéaires et la sélection des fonctions candidates. Les résultats montrent que les deux méthodes identifient efficacement les systèmes testés, mais que l'approche ADAM-SINDy rencontre des limites pour la formulation énergétique dans le cadre du formalisme GENERIC. Nous concluons sur les perspectives d'amélioration et d'application à des données bruitées.

Abstract

This article presents a study on symbolic regression applied to the identification of generic systems. The context focuses on modeling dynamical systems from observed data, with the challenge of discovering interpretable and parsimonious models. We compare the SINDy method and its extension, ADAM-SINDy, which simultaneously optimizes nonlinear parameters and candidate function selection. Results show that both methods efficiently identify the tested systems, but the ADAM-SINDy approach faces limitations for energy formulation within the GENERIC framework. We conclude with perspectives for improvement and application to noisy data.

Contents

1 Introduction

1.1 Overview

System identification is a crucial task in the modeling of dynamical systems. It allows us to study the properties of the system. From a set of observed data, the goal is to find a model that can describe the system's behavior. In our context of machine learning, system identification can be improved by the possibility of modeling complex, nonlinear systems and sometimes uncovering hidden relationships in the data. Recently, techniques such as genetic algorithms and symbolic regression have been developed to generate mathematical models from data with minimal prior hypotheses. By using a limited number of terms in the model, these methods can provide interpretable results.
Symbolic regression is a powerful state-of-the-art tool for discovering mathematical expressions that describe the behavior of complex systems. But one of its major drawbacks is its difficulty in capturing chaotic systems.
Based on the principle that many physical dynamical systems can be described by parsimonious models (in other words, models with a limited number of terms) [2], sparsity-based methods such as the Sparse Identification of Nonlinear Dynamics (SINDy) have been developed to identify such models from data. They allow the discovery of compact, interpretable models by interpreting the problem as a sparse regression problem. However, this technique uses a fixed predefined set of basis functions, which, even though it reduces the computational cost, can limit its ability to capture complex dynamics.
The ADAM-SINDy augmented approach aims to overcome the limitations of SINDy by simultaneously optimizing the nonlinear parameters and selecting candidate functions from a large set of basis functions.
During this internship, we will explore the use of the ADAM-SINDy method to identify mathematical models of generic systems. We will start by describing the general mathematical context in which we want to apply symbolic regression. Then, we will present the SINDy method and its augmented version, ADAM-SINDy. Finally, we will present the results obtained by applying these methods to test functions and then to generic systems.

1.2 Scimba

The main objective of this internship is to implement the SINDy and ADAM-SINDy methods in the Scimba library. Scimba is a Python package developed by the Institut de Recherche Mathématique Avancée (IRMA) for the implementation of different Scientific Machine Learning methods. These are the main features of the package:

- Network implementations: Multi Layer Perceptron (MLP), Discontinuous MLP, RBF networks, activation functions and a basic trainer
- Models of differential equations: Ordinary differential equations (ODE), Partial (PDE), Spatial PDEs, time-space PDEs, etc.
- Specific networks for Physics-Informed Neural Networks (PINN): MLP, Discontinuous MLP, nonlinear RBF networks, Fourier networks, etc.
- Trainer: Each type of PDE has its own trainer

2 Identification Methods

2.1 Mathematical Context

In our context, we are interested in the identification of mathematical models of dynamical systems as follows:

$$\dot{x}(t) = f(x(t)) \tag{1}$$

where $x(t) \in \mathcal{R}^n$ is the state vector of the system at time t, $\dot{x}(t)$ its first time derivative, and $f(x) : \mathcal{R}^n \to \mathcal{R}^n$ is a nonlinear function that describes the dynamics of the system.

2.2 Sparse Identification of Nonlinear Dynamical Systems (SINDy)

To identify nonlinear dynamical systems from a differential equation like (1), we can use the SINDy method [2], which is based on the principle of sparse regression. From a set of observed data:

$$\mathbf{X} = \begin{bmatrix} x(t_1)^T \\ x(t_2)^T \\ \vdots \\ x(t_m)^T \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \tag{2}$$

and a library of candidate functions $\Omega(\mathbf{X}; \mathbf{\Lambda}) = [1\ \mathbf{X}^\alpha\ ...\ sin(\beta\mathbf{X})\ ...\ X \otimes exp(\gamma\mathbf{X})]$ (which can be polynomials, trigonometric functions, exponential, etc), $\otimes$ is the Kronecker product, $\Lambda \in \{\alpha, \beta, \gamma\}$ are the parameters of the functions (respectively the higher polynomial degree, the Fourier frequencies and the exponential decay rate). For now, these parameters can be either fixed which is not ideal, or defined in a grid search, which can be computationally expensive and lead to missing some important terms.
The SINDy method aims to find a sparse coefficient vector $\mathbf{\Theta} = [\theta_1, \theta_2, ..., \theta_p]$ such that:

$$\dot{\mathbf{X}} = \Omega(\mathbf{X}; \mathbf{\Lambda})\mathbf{\Theta} \tag{3}$$

The sparsity of this method is obtained by using a regularization technique such as Lasso augmented with a sequential thresholding, which iteratively solves least squares problems and removes coefficients under a given threshold. The sparse regression problem can be formulated as:

$$\mathbf{\Theta} = \min_{\mathbf{\Theta}} \left(\left\| \dot{\mathbf{X}} - \Omega(\mathbf{X}; \mathbf{\Lambda})\mathbf{\Theta} \right\|_2^2 + \lambda \left\| \mathbf{\Gamma}\mathbf{\Theta} \right\|_1 \right) \tag{4}$$

where λ is a regularization parameter that controls the sparsity of the solution. Generally, in the SINDy method the $\mathbf{\Gamma}$ is the identity matrix, but it can be used to apply a different regularization to each candidate function. And now the SINDy identified dynamical system can be written as:

$$\dot{x}(t) = \Omega(x(t))\mathbf{\Theta} \tag{5}$$

Algorithm 1 SINDy Algorithm

Input: Observational data $\mathbf{X}$ and $\dot{\mathbf{X}}$
  Tolerance ϵ, maximum iterations $N_{\max}$
  Regularization parameter λ
  Sparsity knobs Γ
Initialize: $\Theta \leftarrow \Theta_0$, $\Lambda \leftarrow \Lambda_0$
  $n \leftarrow 0$ ▷ Iteration counter
while $n < N_{\max}$ **do**
  Compute $\Omega(\mathbf{X}; \Lambda)$
  Compute loss:

$$\mathcal{L} \leftarrow \frac{1}{2} \left\| \dot{\mathbf{X}} - \Omega(\mathbf{X}; \Lambda)\Theta \right\|_2^2 + \lambda \|\Gamma\Theta\|_1$$

  Update parameters using backpropagation:

$$\Theta \leftarrow \Theta - \eta_\Theta \nabla_\Theta \mathcal{L}$$

  Update Λ if necessary
  Apply thresholding: For each i, set $\Theta_i, \Lambda_i \leftarrow 0$ if $|\Theta_i|, |\Lambda_i| < \epsilon$
  $n \leftarrow n + 1$
end while
Output: Optimized parameters Θ^* and Λ^*

2.3 Augmented methods ADAM-SINDy

The ADAM-SINDy method [2] is an extension of the SINDy framework that aims to overcome the limitations of the fixed basis functions used in SINDy. It allows the simultaneous optimization of the nonlinear parameters and the selection of candidate functions from a large set of basis functions by replacing the least squares algorithm for the LASSO problem with ADAM optimization. The ADAM-SINDy can be formulated as follows:

$$\mathbf{\Theta} = \min_{\mathbf{\Theta}} \max_{\mathbf{\Gamma} \text{or} \lambda} \left(\left\| \dot{\mathbf{X}} - \Omega(\mathbf{X}; \mathbf{\Lambda})\mathbf{\Theta} \right\|_2^2 + \lambda \left\| \mathbf{\Gamma\Theta} \right\|_1 \right) \tag{6}$$

We minimize the loss function with respect to the coefficients $\mathbf{\Theta}$ and maximize it with respect to the sparsity knobs $\mathbf{\Gamma}$ or λ. We will rather use the sparsity knobs Γ since they control each candidate function's contribution to the model individually.

In algorithm 2, we initialize the parameters Θ, Λ, λ and Γ.
Then, we sample batches of data from the observed data to enhance computational efficiency and robustness, and compute the loss function.
At each epoch, for each batch, we compute the loss function and update the parameters via stochastic gradient descent on a per batch basis or with full-batch. As said before, the sparsity is controlled by the sparsity knobs Γ or λ, which can be updated using a learning rate η_Γ or η_λ.
Finally, we apply a thresholding step to remove the coefficients that are below a given threshold ϵ.

Algorithm 2 ADAM-SINDy Algorithm

Input: Observational data $\mathbf{X}$ and $\dot{\mathbf{X}}$
  Tolerance ϵ, maximum iterations $N_{\max}$
  Learning rates η_Θ, η_Λ, η_λ, η_Γ
Initialize: $\Theta \leftarrow \Theta_0$, $\Lambda \leftarrow \Lambda_0$, $\lambda \leftarrow \lambda_0$, $\Gamma \leftarrow \Gamma_0$
  $n \leftarrow 0$ ▷ Iteration counter
while $n < N_{\max}$ **do**
  for each batch $b = 1, 2, \ldots$ **do**
    Sample batch data $\mathbf{X}_b$, $\dot{\mathbf{X}}_b$
    Compute $\Omega(\mathbf{X}_b; \Lambda)$
    Compute loss:

$$\mathcal{L}_b \leftarrow \frac{1}{2} \left\| \dot{\mathbf{X}}_b - \Omega(\mathbf{X}_b; \Lambda)\Theta \right\|_2^2 + \lambda \|\Gamma\Theta\|_1$$

    Update parameters using backpropagation:

$$\Theta \leftarrow \Theta - \eta_\Theta \nabla_\Theta \mathcal{L}_b$$
$$\Lambda \leftarrow \Lambda - \eta_\Lambda \nabla_\Lambda \mathcal{L}_b$$

    if Uniform Sparsity **then**

$$\lambda \leftarrow \lambda + \eta_\lambda \nabla_\lambda \mathcal{L}_b$$

    else

$$\Gamma \leftarrow \Gamma + \eta_\Gamma \nabla_\Gamma \mathcal{L}_b$$

    end if
    Thresholding: For each i, set $\Theta_i, \Lambda_i \leftarrow 0$ if $|\Theta_i|, |\Lambda_i| \leq \epsilon$
  end for
  $n \leftarrow n + 1$
end while
Output: Optimized parameters Θ^* and Λ^*

3 Generic Formalism

3.1 Mathematical Context

The General Equation for Non-Equilibrium Reversible-Irreversible Coupling (GENERIC) is a mathematical framework that describes the evolution of systems in beyond-equilibrium thermodynamics. It provides a systematic way to model the dynamics of systems with both conservative and dissipative systems. The GENERIC formalism is particularly useful for studying complex systems where energy and entropy exchanges play a crucial role. It can be written as follows [1] [3]:

$$\begin{cases} \dot{x}(t) = L(x(t))\,\nabla E(x(t)) + M(x(t))\,\nabla S(x(t)) \\ L\nabla S = 0 \\ M\nabla E = 0 \end{cases} \tag{7}$$

where $L\nabla E$ is the conservative part of the system, $M\nabla S$ is the dissipative part, and E and S are both the Energy and the Entropy of the system, respectively. L is the skew-symmetric Poisson Matrix and M is the symmetric semi-definite friction matrix. This system checks that the total energy of the system is conserved, while the entropy is monotonically increasing.

3.2 Modified algorithm

The main idea is to add to existing Adam-SINDy's method the creation of a functions' library specially for the Energy and then to the Loss a term that will impose that $M\nabla E = 0$ and $L\nabla S = 0$ when minimizing . To do so, I chose to compute a mean square error (mse) between each of these two terms and 0 thus ensuring the convergence. The optimization parameters for this part of the total loss are the coefficients of the matrices G and J.
L is chosen to be :

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Algorithm 3 ADAM-SINDy / Generic Formalism Algorithm

Input: Observational data $\mathbf{X}$ and $\dot{\mathbf{X}}$
Skew-symmetric matrix $\mathbf{L}$ and the symmetric semi-definite $\mathbf{M}$
Tolerance ϵ, maximum iterations $N_{\max}$
Learning rates η_Θ, η_Λ, η_λ, η_Γ, η_M
Initialize: $\Theta \leftarrow \Theta_0$, $\Lambda \leftarrow \Lambda_0$, $\lambda \leftarrow \lambda_0$, $\Gamma \leftarrow \Gamma_0$
$n \leftarrow 0$ ▷ Iteration counter
while $n < N_{\max}$ **do**
  for each batch $b = 1, 2, \ldots$ **do**
    Sample batch data $\mathbf{X}_b$, $\dot{\mathbf{X}}_b$
    Compute $\Omega(\mathbf{X}_b; \Lambda)$, $\Omega_E(\mathbf{X}_b; \Lambda)$
    Compute loss:

$$\mathcal{L}_b \leftarrow \frac{1}{2}\left\|\dot{\mathbf{X}}_b - \Omega(\mathbf{X}_b;\Lambda)\Theta\right\|_2^2 + \lambda\|\Gamma\Theta\|_1 + \frac{1}{2}\|\mathbf{M}\nabla\mathbf{E}_b\|_2^2 + \frac{1}{2}\|\mathbf{G}\nabla\mathbf{S}_b\|_2^2$$

    Update parameters using backpropagation:

$$\Theta \leftarrow \Theta - \eta_\Theta \nabla_\Theta \mathcal{L}_b$$
$$\Lambda \leftarrow \Lambda - \eta_\Lambda \nabla_\Lambda \mathcal{L}_b$$

    if Uniform Sparsity **then**

$$\lambda \leftarrow \lambda + \eta_\lambda \nabla_\lambda \mathcal{L}_b$$

    else

$$\Gamma \leftarrow \Gamma + \eta_\Gamma \nabla_\Gamma \mathcal{L}_b$$

    end if

$$M \leftarrow M - \eta_M \nabla_M \mathcal{L}_b$$

    Thresholding: For each i, set $\Theta_i, \Lambda_i \leftarrow 0$ if $|\Theta_i|, |\Lambda_i| \leq \epsilon$
  end for
  $n \leftarrow n + 1$
end while
Output: Optimized parameters Θ^* and Λ^*

4 Results

4.1 Problem Statement

SINDy's method is commonly built upon the master functions library :

$$\Omega(\mathbf{X};\mathbf{\Lambda}) = [1 \;\; \mathbf{X^A} \;\; sin(B\mathbf{X}) \;\; cos(C\mathbf{X}) \;\; exp(D\mathbf{X}) \;\; \mathbf{X} \otimes sin(E\mathbf{X}) \;\; \mathbf{X} \otimes cos(F\mathbf{X}) \;\; \mathbf{X} \otimes exp(G\mathbf{X})] \quad (9)$$

where $\mathbf{X}$ are the input data, A, B, C, D, E, F, G are the adaptative non-linear parameters (Λ) of the functions that are imposed for the Sindy's method and will be optimized for the Adam Sindy's method except for the coefficient A that we will force to be 1. The coefficient B, C, E, F which are the fourier frequencies of the problem will be initialized at 1 and D, G, the exponential decay rate, at 0.1. These trainable parameters add another advantage to the method since for values of C and D close to 0 the cosine and exponential interact with the term 1. And for values of F and G close to 0, the associated terms interact with the term $\mathbf{X}^A$. This allows the method to expand the library of candidate functions and to capture more complex dynamics.

4.2 Presentation of the examples

To test the implemented methods, we will use the harmonic oscillator [2] and the damped nonlinear oscillator [1] as examples described by the following systems of equations:

Harmonic Oscillator

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -x_1(t) + 0.1\, x_2(t)\, cos(0.75\, x_1(t)) \end{cases} \quad (10)$$

Damped nonlinear Oscillator

$$\begin{cases} \dot{q}(t) = p(t) \\ \dot{p}(t) = -3\, sin(q(t)) - 0.04\, p(t) \\ \dot{S}(t) = -0.04\, p(t)^2 \end{cases} \quad (11)$$

The equation of the energy of the system is given by:

$$E(t) = 0.5\, p(t)^2 - 3\, \cos(q(t)) + S \quad (12)$$

4.3 SINDy Results

4.3.1 Harmonic Oscillator

For the harmonic oscillator, I use the following libraries of candidate functions:

- $\Omega_1 = [x_2]$
- $\Omega_2 = [x_1, x_2 \otimes cos(0.75\, x_1)]$

For 50000 iterations, with tmax = 50s, dt 0.01, $\lambda = 0.001$, an initial learning rate of 0.1, the SINDy method identifies the following model:

$$\begin{cases} \dot{x}_1(t) = 1.00199711322784\, x_2 \\ \dot{x}_2(t) = -0.999280571937561\, x_1 + 0.101232923567295\, x_2 \cos(0.75 x_1) \end{cases} \quad (13)$$

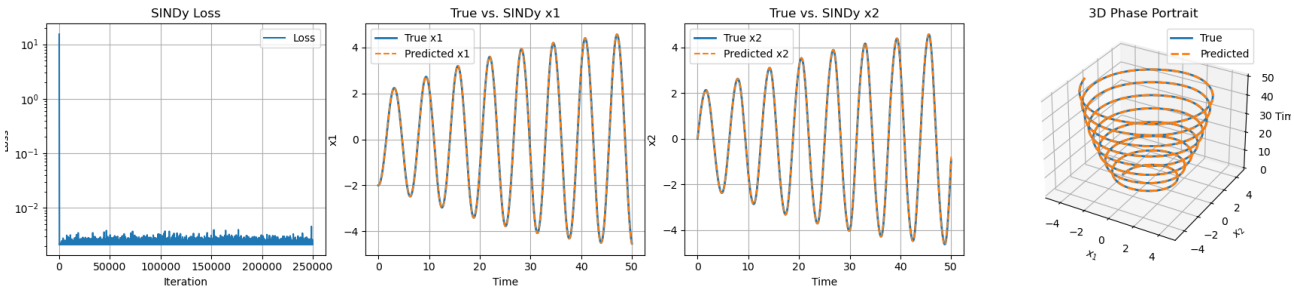

Figure 1: Identified model for the harmonic oscillator

The identified model is very close to the original system, with a small error of 10^{-3}.

4.3.2 Damped nonlinear Oscillator

For the damped nonlinear oscillator, I use the following libraries of candidate functions:

- $\Omega_1 = [p]$
- $\Omega_2 = [p, sin(q)]$
- $\Omega_3 = [p^2]$

For 50000 iterations, with tmax = 5s, dt 0.001, $\lambda = 0.001$, an initial learning rate of 0.001, the SINDy method identifies the following model:

$$\begin{cases} \dot{q}(t) = 0.999506831169128\, p(t) \\ \dot{p}(t) = -0.0397274941205978\, p(t) + -3.00022983551025\, \sin(q(t)) \\ \dot{S}(t) = 0.0395135618746281\, p(t)^2 \end{cases} \tag{14}$$

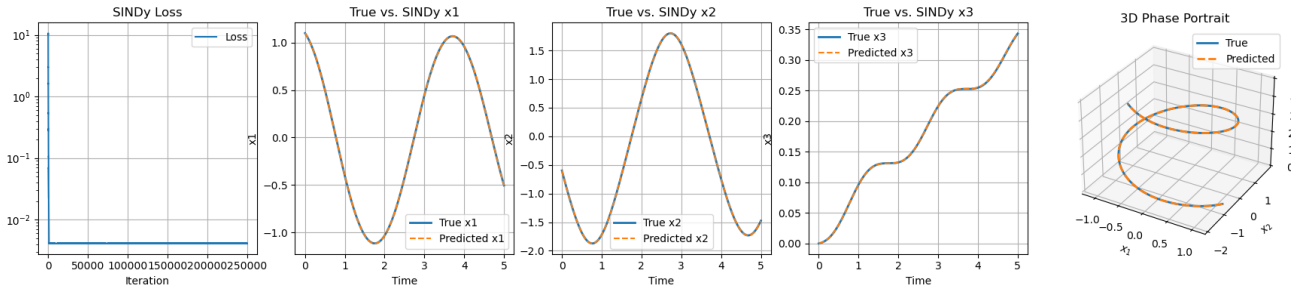

Figure 2: Identified model for the damped nonlinear oscillator

Again, the identified model is very close to the original system, with a small error of 10^{-3}. For the energy of the system, we have:

$$E(t) = 0.500103414058685\, p(t)^2 + 1.0\, S(t) - 2.99728417396545\, \cos(1.0\, q(t)) \tag{15}$$

Let us note that the Entropy term has to be added manually since it can't be identified during the computation of the symbolic method for this example. But for the rest, the linear and nonlinear coefficients are very close to the theoritical system with an error of 10^{-3}.

4.4 ADAM-SINDy Results

4.4.1 Harmonic Oscillator

For 50000 iterations, with tmax = 50s, dt 0.001, $\lambda = 0.001$, an initial learning rate of 0.01 and a pruning coefficient $\epsilon = 0.005$ the ADAM-SINDy method identifies the following model:

$$\begin{cases} \dot{x}_1(t) = 1.00006556510925\, x_2 + 0.000205039978027344\, x_1 \\ \dot{x}_2(t) = -1.0004680454731\, x_1 + 0.100180670619011\, x_2 \cos(0.75 x_1) \end{cases} \tag{16}$$

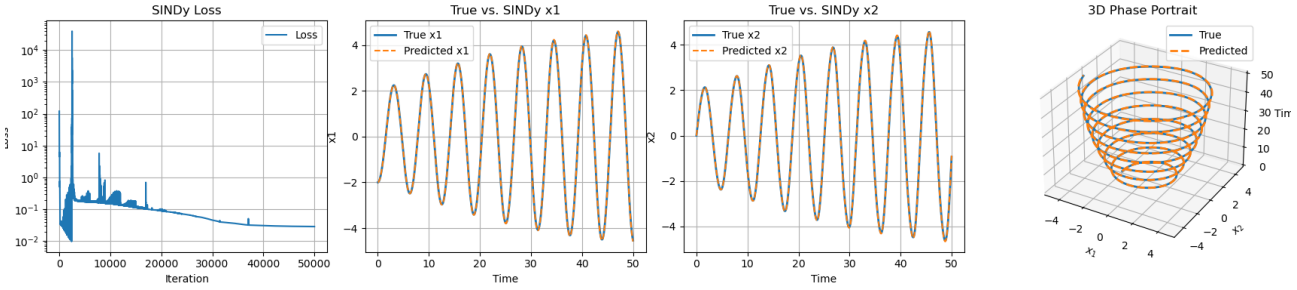

Figure 3: Identified model for the harmonic oscillator with ADAM-SINDy

The identified model is very close to the original system, with errors from 10^{-3} to 10^{-5}. The Adam-SINDy method is able to succesfully identify the nonlinear parameter of the system.

4.4.2 Damped nonlinear Oscillator

For 50000 iterations, with tmax = 5s, dt 0.001, $\lambda = 0.001$, an initial learning rate of 0.01 and a pruning coefficient $\epsilon = 0.005$ the ADAM-SINDy method identifies the following model:

$$\begin{cases} \dot{q}(t) = 0.999236464500427\, p(t) \\ \dot{p}(t) = -1.98613214492798\, p(t)\, \exp(0.01\, S(t)) - 0.923614144325256\, p(t)\, \cos(0.92\, p(t)) \\ \qquad + 0.0422132685780525\, p(t)\, \cos(1.82\, S(t)) - 0.0287085622549057\, S(t)\, \exp(-0.9\, p(t)) \\ \dot{S}(t) = 0.0395686700940132\, p(t)^2 \end{cases} \tag{17}$$

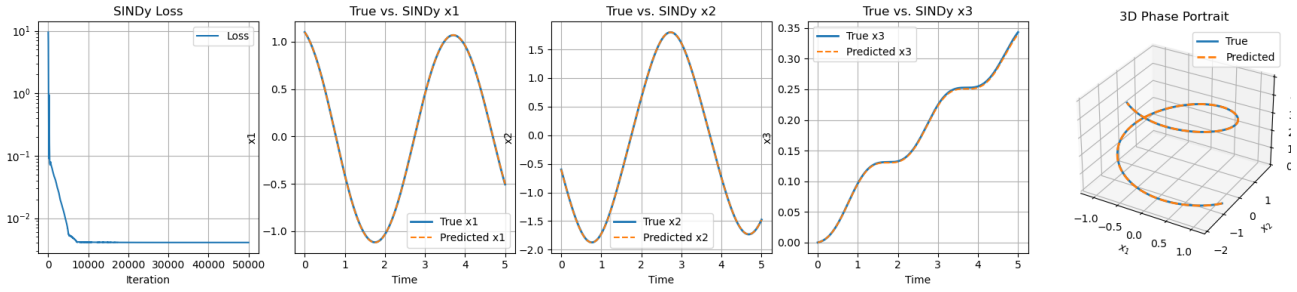

Figure 4: Identified model for the damped nonlinear oscillator with ADAM-SINDy

Once again, the identified model is very close to the original system, but for the symbolic formula of $p(t)$ we can observe terms that shouldn't appear. However, the model is able to capture the main dynamics of the system.

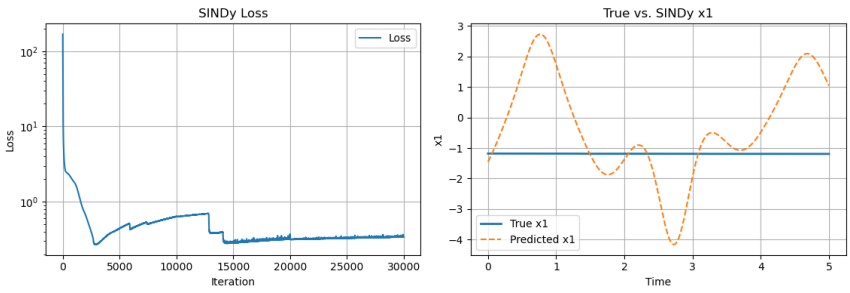

Figure 5: Identified Energy formula for the damped nonlinear oscillator with ADAM-SINDy

In this case, the GENERIC formalism doesn't seem to be able to capture the right behaviour for the energy. The ADAM-SINDy method fail to identify the energy of the system.

5 Conclusion

I have implemented both SINDy and augmented SINDy (ADAM-SINDy) methods in the SCIMBA library and extended them with the structure-preserving parametrization, the GENERIC formalism. I tested my implementation with a benchmark of examples.

For both methods, the identification of the dynamical system was a success, but for the association with the GENERIC formalism, only the SINDy method proved to be working, whereas the ADAM-SINDy method showed some limitations when identifying both the behaviour and the symbolic formulation of the problem. It will be interesting to investigate the problem. It might be a simple issue of choosing the right initial parameters or maybe a structural issue.

It would also be very interesting to add the treatment of noise in the case of real signals to study. Such methods are already implemented in SCIMBA, but mixing them with the presented work remains to be done.

And of course, these methods could be optimized, indeed the computation of the previous results took between 350s to more than 500s.

Finally, I would like to thank my supervisors, Dr. Emmanuel Franck, Dr Andrea Thomann and Dr Michel Victor-Dansac, for their help and support during this project.

References

[1] Kookjin Lee, Nathaniel Trask, and Panos Stinis. "Structure-preserving Sparse Identification of Nonlinear Dynamics for Data-driven Modeling". In: (2021).

[2] Siva Viknesh, Younes Tatari, and Amirhossein Arzani. "ADAM-SINDy: An Efficient Optimization Framework for Parameterized Nonlinear Dynamical System Identification". In: (2025).

[3] Zhen Zhang, Yeonjong Shin, and George Em Karniadakis. "GFINNs: GENERIC Formalism Informed Neural Networks for Deterministic and Stochastic Dynamical Systems". In: (2021).