



**Universidade do Estado do Rio de Janeiro**

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Diego Matos Silva Lopes


**EVALUATION OF A SPARSE REGRESSION MACHINE  
LEARNING TECHNIQUE FOR DYNAMICAL SYSTEMS  
DISCOVERY**

Rio de Janeiro

2024

Diego Matos Silva Lopes

**EVALUATION OF A SPARSE REGRESSION MACHINE LEARNING  
TECHNIQUE FOR DYNAMICAL SYSTEMS DISCOVERY**



Master's Thesis presented to the Mechanical Engineering Graduate Program of the Universidade do Estado do Rio de Janeiro as a partial requirement to obtain the degree of Master in Sciences. Field of concentration: Solid Mechanics.

Advisor: Prof. Americo Barbosa da Cunha Junior, D.Sc.

Rio de Janeiro

2024

CATALOGAÇÃO NA FONTE  
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

L864    Lopes, Diego Matos Silva.  
          Evaluation of a sparse regression machine learning technique for  
          dynamical systems discovery / Diego Matos Silva Lopes. – 2024.  
          78 f.

          Orientador: Americo Barbosa da Cunha Junior.  
          Dissertação (Mestrado) – Universidade do Estado do Rio de  
          Janeiro, Faculdade de Engenharia.

          1. Engenharia mecânica - Teses. 2. Aprendizado do computador -  
          Teses. 3. Sistemas dinâmicos diferenciais - Teses. I. Cunha Junior,  
          Americo Barbosa da. II. Universidade do Estado do Rio de Janeiro,  
          Faculdade de Engenharia. III. Título.

CDU 517.93

Bibliotecária: Júlia Vieira – CRB7/6022

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou  
parcial desta tese, desde que citada a fonte.

---

Assinatura

01/07/2024

---

Data

Diego Matos Silva Lopes

**EVALUATION OF A SPARSE REGRESSION MACHINE LEARNING  
TECHNIQUE FOR DYNAMICAL SYSTEMS DISCOVERY**

Master's Thesis presented to the Mechanical Engineering Graduate Program of the Universidade do Estado do Rio de Janeiro as a partial requirement to obtain the degree of Master in Sciences. Field of concentration: Solid Mechanics.

Approved on march 08, 2024.

Examining Committee:

---

Prof. Americo Barbosa da Cunha Junior, D.Sc. (Advisor)  
Universidade do Estado do Rio de Janeiro (UERJ)

---

Prof. Fernando Alves Rochinha, D.Sc.  
Universidade Federal do Rio de Janeiro (UFRJ)

---

Prof. Karla Tereza Figueiredo Leite, D.Sc.  
Universidade do Estado do Rio de Janeiro (UERJ)

---

Prof. Samuel da Silva, D.Sc.  
Universidade Estadual Paulista (UNESP)

Rio de Janeiro

2024

## DEDICATION

I dedicate this to my late grandmother Adinilsa de Matos Silva; Semper in memoria mea.

## ACKNOWLEDGMENTS

I would like to express my deepest appreciation to all my closest family members and all the integrants of NUMERICO reseacrh group for the support and help, especially for my mother Wanda Matos, my advisor Prof. Americo Cunha Jr. And express my appreciation to the FAPERJ for funding my master's scholarship under the following grant 200.525/2020.

## ABSTRACT

MATOS SILVA LOPES, Diego. *Evaluation of a Sparse Regression Machine Learning Technique for Dynamical Systems Discovery*. 2024. 78 f. Master's Thesis (Master in Mechanical Engineering) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil, 2024.

This work comprehensively and didactically presents the Sparse Identification of Nonlinear Dynamics (SINDy) method, applying it to various dynamic systems to assess the robustness and effectiveness of the method in inferring their evolution equations. The method proved effective in inferring dynamic systems, including chaotic ones. A test with a trigonometric nonlinearity using only polynomial functions in the candidate function library was conducted. It was observed that the Taylor series polynomials for this function were inferred, albeit with differences in parameters and the presence of a dissipative term. Furthermore, a sequence of tests was developed, where two crucial statistical parameters (root mean square error and correlation) were calculated to evaluate the quality of the inferred dynamics compared to the dynamics that originated the data. The quality of the obtained data was altered in three ways: an increased number of data samples in the same time interval, different intensities of Gaussian noise in the data, and varying time intervals while maintaining the same number of data points. In each of these tests, the number of candidate functions was also altered to assess the influence of additional functions on result quality. The results showed a significant impact of the number of data points and noise on the outcomes. Conversely, the data capture interval did not yield differences in the measured values to evaluate the proximity of dynamics between different intervals. With an increased number of functions, there was a tendency for divergence between the dynamics.

Keywords: Dynamical systems; Nonlinear dynamics; Machine learning; SINDy.

## RESUMO

MATOS SILVA LOPES, Diego. *Avaliação de uma Técnica de Aprendizado de Máquina de Regressão Esparsa para Identificar Sistemas Dinâmicos*. 2024. 78 f. Master's Thesis (Master in Mechanical Engineering) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil, 2024.

Este trabalho apresenta de forma didática e completa o método de SINDy, aplicando-o em diferentes sistemas dinâmicos em busca de testar a robustez e eficácia do método em inferir sistemas caóticos, onde o método mostrou-se eficaz em inferir os sistemas dinâmicos, até mesmo os caóticos. Um teste com um sistema trigonométrico apenas com funções polinomiais na biblioteca de funções candidatas foi realizado, onde foi observado que os polinômios da série de Taylor para esta função foi inferido mas com diferença nos parâmetros e com a presença de um termo dissipativo. Também foi desenvolvido uma sequência de testes onde dois importantes parâmetros estatísticos (raiz quadrada do erro médio e correlação) são calculados para avaliar a qualidade da dinâmica inferida em relação a dinâmica que deu origem aos dados ao alterar a qualidade dos dados obtidos, três diferentes alterações na qualidade dos dados são propostos, maior número de amostra de dados em um mesmo intervalo de tempo, diferentes intensidades de ruído gaussiano nos dados e por último diferentes intervalos de tempo mantendo o mesmo número de amostras de pontos de dados, em cada um destes testes também foi alterado o número de funções candidatas para avaliar a influência que funções a mais causam na qualidade do resultado. Observou-se que o número de pontos de dados e o ruído afetaram consideravelmente os resultados, enquanto que o intervalo de captura dos pontos de dados não apresentou diferença nos valores de medidos para avaliar a proximidade das dinâmicas entre os diferentes intervalos. Quanto maior o número de funções houve uma tendência de aumento de divergência entre as dinâmicas.

Palavras-chave: Sistema dinâmicos; Dinâmica não linear; Aprendizado de máquinas; SINDy.



## LIST OF FIGURES

Figure 1 - Overfitting . . . . .	15
Figure 2 - SINDy schematics . . . . .	23
Figure 3 - Benchmarks systems . . . . .	25
Figure 4 - Generated data . . . . .	29
Figure 5 - Duffing oscillator numerical dynamics compared with data driven dynamics . . . . .	30
Figure 6 - Differences in numerical dynamics versus data-driven approaches over extended durations . . . . .	31
Figure 7 - Duffing Oscillator Numerical Dynamics Compared with Data-Driven Training with Different Initial Conditions . . . . .	32
Figure 8 - Comparison between the chaotic Duffing oscillator's numerical dynamics and its data-driven dynamics . . . . .	34
Figure 9 - Time series of the simple pendulum with a Taylor series . . . . .	36
Figure 10 - Van der Pol oscillator numerical dynamics compared with data-driven dynamics . . . . .	37
Figure 11 - Three-dimensional phase space of the Van der Pol oscillator . . . . .	38
Figure 12 - Time Series of the Three Dimensions of the Rössler System . . . . .	39
Figure 13 - Rössler Attractor That Originated the Data Compared to the One Reconstructed by the Identified Dynamics . . . . .	40
Figure 14 - Average RMSE for Different Numbers of Time Series . . . . .	42
Figure 15 - RMSE with different number of data points . . . . .	44
Figure 16 - Correlation of $\dot{x}_1$ with different number of data points . . . . .	46
Figure 17 - Correlation of $\dot{x}_2$ with different number of data points . . . . .	48
Figure 18 - Correlation of $\dot{x}_3$ with different number of data points . . . . .	49
Figure 19 - Correlation of phase space with different number of data points . . . . .	51
Figure 20 - RMSE with different intensity noise . . . . .	54
Figure 21 - Correlation of $\dot{x}_1$ with different intensity noise . . . . .	56
Figure 22 - Correlation of $\dot{x}_2$ with different intensity noise . . . . .	58
Figure 23 - Correlation of $\dot{x}_3$ with different intensity noise . . . . .	59
Figure 24 - Correlation of phase space with different intensity noise . . . . .	61
Figure 25 - RMSE with different time interval . . . . .	63
Figure 26 - Correlation of $\dot{x}_1$ with different time interval . . . . .	65
Figure 27 - Correlation of $\dot{x}_2$ with different time interval . . . . .	67
Figure 28 - Correlation of $\dot{x}_3$ with different time interval . . . . .	68
Figure 29 - Correlation of phase space with different time interval . . . . .	70

## LIST OF TABLES

Table	1 - RMSE for different number of data points with poly order 3 . . . . .	43
Table	2 - RMSE for different number of data points with poly order 4 . . . . .	43
Table	3 - RMSE for different number of data points with poly order 5 . . . . .	45
Table	4 - Correlation $\dot{x}_1$ for different number data points with poly order 3 . . .	45
Table	5 - Correlation $\dot{x}_1$ for different number data points with poly order 4 . . .	47
Table	6 - Correlation $\dot{x}_1$ for different number data points with poly order 5 . . .	47
Table	7 - Correlation $\dot{x}_2$ for different number data points with poly order 3 . . .	47
Table	8 - Correlation $\dot{x}_2$ for different number data points with poly order 4 . . .	47
Table	9 - Correlation $\dot{x}_2$ for different number data points with poly order 5 . . .	47
Table	10 - Correlation $\dot{x}_3$ for different number data points with poly order 3 . . .	47
Table	11 - Correlation $\dot{x}_3$ for different number data points with poly order 4 . . .	50
Table	12 - Correlation $\dot{x}_3$ for different number data points with poly order 5 . . .	50
Table	13 - Total correlation for different number of data points with poly order 3	50
Table	14 - Total correlation for different number of data points with poly order 4	50
Table	15 - Total correlation for different number of data points with poly order 5	50
Table	16 - RMSE for different noise intensity with poly order 3 . . . . .	53
Table	17 - RMSE for different noise intensity with poly order 4 . . . . .	53
Table	18 - RMSE for different noise intensity with poly order 5 . . . . .	55
Table	19 - Correlation of $\dot{x}_1$ for different noise intensity with poly order 3 . . . . .	55
Table	20 - Correlation of $\dot{x}_1$ for different noise intensity with poly order 4 . . . . .	57
Table	21 - Correlation of $\dot{x}_1$ for different noise intensity with poly order 5 . . . . .	57
Table	22 - Correlation of $\dot{x}_2$ for different noise intensity with poly order 3 . . . . .	57
Table	23 - Correlation of $\dot{x}_2$ for different noise intensity with poly order 4 . . . . .	57
Table	24 - Correlation of $\dot{x}_2$ for different noise intensity with poly order 5 . . . . .	57
Table	25 - Correlation of $\dot{x}_3$ for different noise intensity with poly order 3 . . . . .	57
Table	26 - Correlation of $\dot{x}_3$ for different noise intensity with poly order 4 . . . . .	60
Table	27 - Correlation of $\dot{x}_3$ for different noise intensity with poly order 5 . . . . .	60
Table	28 - Total correlation for different noise intensity with poly order 3 . . . . .	60
Table	29 - Total correlation for different noise intensity with poly order 4 . . . . .	60
Table	30 - Total correlation for different noise intensity with poly order 5 . . . . .	60
Table	31 - RMSE for different time interval with poly order 3 . . . . .	62
Table	32 - RMSE for different time interval with poly order 4 . . . . .	64
Table	33 - RMSE for different time interval with poly order 5 . . . . .	64
Table	34 - Correlation of $\dot{x}_1$ for different time interval with poly order 3 . . . . .	64
Table	35 - Correlation of $\dot{x}_1$ for different time interval with poly order 4 . . . . .	66
Table	36 - Correlation of $\dot{x}_1$ for different time interval with poly order 5 . . . . .	66

Table 37 - Correlation of $\dot{x}_2$ for different time interval with poly order 3 . . . . .	66
Table 38 - Correlation of $\dot{x}_2$ for different time interval with poly order 4 . . . . .	66
Table 39 - Correlation of $\dot{x}_2$ for different time interval with poly order 5 . . . . .	66
Table 40 - Correlation of $\dot{x}_3$ for different time interval with poly order 3 . . . . .	66
Table 41 - Correlation of $\dot{x}_3$ for different time interval with poly order 4 . . . . .	69
Table 42 - Correlation of $\dot{x}_3$ for different time interval with poly order 5 . . . . .	69
Table 43 - Total correlation for different time interval with poly order 3 . . . . .	69
Table 44 - Total correlation for different time interval with poly order 4 . . . . .	69
Table 45 - Total correlation for different time interval with poly order 5 . . . . .	69

## SUMMARY

	<b>GENERAL INTRODUCTION</b>	11
1	<b>SPARSE IDENTIFICATION OF NONLINEAR DYNAMICS</b>	18
2	<b>BENCHMARKS SYSTEMS</b>	24
2.1	Duffing oscillator	24
2.2	Simple pendulum	26
2.3	Van der Pol Oscillator	26
2.4	Rössler System	27
3	<b>RESULTS AND DISCUSSION</b>	28
3.1	Generated Data	28
3.2	Analysis of the Duffing oscillator	28
3.3	Analysis of the pendulum dynamics	33
3.4	Analysis of the Van der Pol oscillator	35
3.5	Analysis of the Rössler system	38
3.6	Convergence Test for Number of Simulations	41
3.7	Analysis of the RMSE and Correlation	41
3.7.1	<u>Number of data points</u>	43
3.7.2	<u>Noise intensity</u>	52
3.7.3	<u>Time interval</u>	62
	<b>CONCLUSIONS</b>	71
	<b>REFERENCES</b>	73

## GENERAL INTRODUCTION

The majority of classical physics, mathematical, biological equations, among other fields of science, has been discovered thru intense work of observation, experiments, and usage of first principles to reach an equation that describes the phenomenon studied. These first principles include phenomena like the balance of mass, Newton's law of motion, the laws of thermodynamics, and others (ODEN, 2011). However, for modern dynamical systems where these first principles are unknown, some examples are epidemiological modeling (DANTAS; TOSIN; Cunha Jr, 2018; DANTAS; TOSIN; Cunha Jr, 2019; RITTO; JR; BARTON, 2021; LOPES; Cunha Jr, 2022; JR; BARTON; RITTO, 2023), structural health monitoring (VILLANI; SILVA; Cunha Jr, 2018; VILLANI et al., 2019; YANO et al., 2023), neuroscience (GLASER et al., 2019; MARBLESTONE; WAYNE; KORDING, 2016; RICHARDS; LILICRAP; BEAUDOIN, 2019). Discovering these dynamic systems' evolution law becomes an almost impossible assignment to complete analytically.

With the advent of the information age and the evolution of computers, numerical methods, and artificial intelligence (AI), new ways of studying, working, analyze is being developed every single day, revolutionizing science. Some specialists in economics agree that we are in the fourth industrial revolution, works of Schwab (2016), Xu, David and Kim (2018) show the evolution between each industrial revolution and highlight the challenges and opportunities of the actual revolution. These works also explore new tools that arise in this new era. Examples of these are the internet of things (IoT), advanced robotics, 3D printing, and cognitive computing. Nevertheless, another essential term is big data. Never before in the human story was created and stored the quantity of data like now.

In recent years, the evolution of AI, especially machine learning (ML) techniques, has been impressive. All this started with the study of Rosenblatt (1958) leads to the perceptron's creation in the late '50s. The creation of the nearest neighbor algorithm in the work of Cover and Hart (1967) in the '60s. After that, all the research in AI and ML was essential to develop all the tools used today, tools like multilayers neural networks, feedforward neural networks, and backpropagation, regression techniques such as LASSO (Least absolute shrinkage and selection operator) and Bayesian linear regression.

The neural networks (NN) was not very popular in the '70s and '80s, but with the increase of computational processing power and improvements in software and programming, NN became popular in the '90s. The successful results in speech recognition by Hochreiter and Schmidhuber (1997) is one example of the applications done for the NN in the past. Nowadays, the applications are countless, including product recommendations, natural language procession, fraud detection, and more. All this is possible thanks to the amount of data available today. In the scientific mean, that is not different. It

works like (RICHARDS; LILLICRAP; BEAUDOIN, 2019; HASSANIBESHELI; BOERS; KURTHS, 2020; LECUN; BENGIO; HINTON, 2015; MILLS; SPANNER; TAMBLYN, 2017; PATHAK et al., 2018; ZIO; ROCHINHA, 2020) developing NN, in the fields of kernel methods; some examples are (CAMASTRA; VERRI, 2005; APSEMIDIS; PSARAKIS; MOGUERZA, 2020; BADDOO et al., 2022), and even more, complex neural networks like Physics-informed neural networks (ZHANG; GUO; KARNIADAKIS, 2019; ALMEIDA; SILVA; JR, 2023; NATH et al., 2023; HE; ZHAO; YAN, 2023; ZHANG et al., 2024) use to recognize complex patterns based on physics. Despite having an incredible generalization, NN-based methods lack in providing interpretability to the user. Therefore, regression methods are more proper for providing the interpretability that NN does not have.

Nonlinear equations are not rare for complex engineering problems, e.g., the simple pendulum, Duffing oscillator, Navier-Stokes, and others. Typically, complex dynamical systems undergo simplifications to enable mathematical modeling. Thus, utilizing data acquired from sensors of the phenomenon of interest has the benefit of obtaining more reliable data reflecting reality. Nevertheless, it is more and more common to use ML techniques in engineering problems. That varies in helping control complex systems (VAMVOUDAKIS et al., 2015), predict mechanical failure (LI et al., 2014), helping with the engineering design (PANCHAL et al., 2019), and others. However, with the evolution of sensors and the amount of data available today, it is not impossible to use data to discover the evolution law that governs this data set. Some advances in regression methods are getting prominence results for nonlinear dynamical systems. The works of Brunton, Proctor and Kutz (2016) show a new method able to deal with nonlinear dynamical systems with only data.

## Objective

This study aims to investigate the capabilities of a relatively novel ML technique, known as Sparse Identification of Nonlinear Dynamics (SINDy), in the complex task of inferring nonlinear evolution equations from time series data. The goal is to get an insight into the method's strong and weak points and construct pedagogical material for future generations of students who may be interested in applying this technique to nonlinear dynamics problems. This document can be seen as a shortcut to those interested in SINDy. Initially, we offer a comprehensive overview of the methodology, followed by a series of tests encompassing a variety of dynamic systems. These tests include scenarios characterized by both regular and chaotic behavior across different dynamical systems. Additionally, we assess the method's capability to interpret trigonometric dynamics solely through polynomial functions, illustrated using the simple pendulum dynamics.

Furthermore, a set of tests is conducted by systematically varying the hyperparam-

eters of the ML technique to infer their practical effect. Through a statistical analysis, we investigate how the quality of these parameters impacts the method's overall performance. The objective is to identify which simulation parameters exert a more pronounced influence on the results and quantify this impact.

## Literature Review

The field of ML evolved considerably in the past few years, but the essence is unknown to most people. Samuel (1959) wrote in his paper: *"... a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program"*. In this same work, the author complements: *"Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort"*. The significant difference of ML with traditional programming is that with ML algorithms, the computer can use data, trials, or both to learning a specific task without the necessity of explicitly wrote a code that contemplates all possible incomes. Furthermore, is used a scoring method to measure how good the computer is for these tasks.

This definition contemplates many methods that are very different from each other. With that, some authors of ML create subgroups of specific characteristics to distinguish the methods. The author Géron (2019) classified them depending on specific criteria, and the most important and widely used is the training method is supervised with humans or not. The other two are whether or not the method can learn on the fly, and the last, whether the technique work by comparing a new data point with the training data set or detecting pattern on the training data to construct a predictive model.

The principal classification based on human supervision or not has three or four subclassifications. Authors like Bishop (2006) and Herbrich (2002) divided this into supervised, unsupervised, and reinforcement learning sub-categories. The first has the name supervised because the data training has labels, and the ML method can predict or classify based on these labels. The unsupervised does not have labels on the training data, and the method tries to without this information to find patterns. The last is very different from the other two; the ML algorithm observes the result obtained after the training, and by a set of parameters chosen by the programmer is obtained a score, after the following run of the code with, any positive change in the outcome results in a better score, and the opposite is true. More recent works like Géron (2019) considers one more, semi-supervised learning. These methods can deal with data sets data with labels mixed with unlabeled data.

The regression methods like linear, polynomial, and logistic regression are in the supervised learning category. However, regression techniques do not start with ML. Re-

gression is a vital tool in statistical modeling. The first and earliest form of regression is the classical method of least squares developed in the XIX century by Legendre and Gauss to solve problems in astronomy, much time before any computer.

The basic premise of the regression method is to estimate the relationship between the variables that minimize the error of the predicted model to the data. The most basic regression is the simple linear regression, given by

$$y = \beta_0 + \beta_1 x + \epsilon, \quad (1)$$

where  $y$  and  $x$  are the dependent and independent variables, respectively, the  $\beta_0$  and  $\beta_1$  are the coefficients that the method seeks to discover, and the  $\epsilon$  is the error term that the method tries to minimize. The standard approach to do this is to use the method of the least square to obtain the values of  $\beta_0$  and  $\beta_1$  that best fit the data. The first step is to rewrite equation (1), isolating the error or residual  $\epsilon$  resulting in

$$\epsilon = y - (\beta_0 + \beta_1 x), \quad (2)$$

as the objective of the least square is to minimize the sum of squared residuals, or the mean squared error (MSE), given by

$$\text{MSE} = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2, \quad (3)$$

where  $n$  is the number of data points.

To obtain the minimal value is necessary to take the partial derivative of MSE with the coefficients and obtain the value of the result equation equal to zero. The result equations are

$$\frac{\partial \text{MSE}}{\partial \beta_0} = 2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)(-1) = 0 \quad (4)$$

and

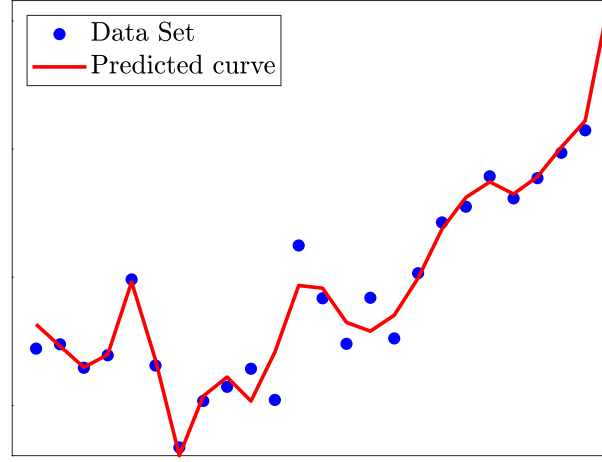
$$\frac{\partial \text{MSE}}{\partial \beta_1} = 2 \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)(-x_i) = 0. \quad (5)$$

Working in these equations, the resulting expression for the estimated coefficients' are

$$\hat{\beta}_1 = \frac{\sum x_i y_i - (\sum x_i)(\sum y_i)/n}{\sum x_i^2 - (\sum x_i)^2/n} \quad (6)$$



Figure 1 - Overfitting



Caption: An example of overfitting in a simple data set, it is possible to notice how the predicted curve is adjusting to the fluctuations of the data set.

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \quad (7)$$

where  $\bar{x}$  and  $\bar{y}$  are the mean values of  $x$  and  $y$ , respectively. For more details of the arithmetics and mathematical properties, the work of Ryan (2008) brings all the details.

After that, more complex regression formulas with this basic premise were created, e.g., the polynomial or logistic regression, where high order terms are present to capture more information. However, one of the biggest problems with this method is the possibility of overfitting the data; i.e., the predicted equation is adjusting too much with the data fluctuations. Figure 1 shows an example of overfitting. The predicted curve follows the fluctuation of the data set, resulting in problems of prediction. However, overfitting is easy to detect for simple problems with low dimensional data, but that is not trivial for a more complex data set.

One way to avoid overfitting is using regularized regression, where a regularized parameter lambda is present in error. This parameter penalizes the error term intending to decrease the number of independent variables of the predicted curve. Using a  $\lambda$  term equal to zero is equivalent to the regression with the least-squares method. When  $\lambda$  equal to infinity, all variables will decrease to zero. The two most famous regularized regressions are Ridge and Lasso regression (HOERL; KENNARD, 1970; TIBSHIRANI, 1996). The

Ridge regression or L2 regularization will penalize the method as follows

$$J = \text{MSE} + \lambda \sum_{j=1}^p \beta_{1j}^2, \quad (8)$$

where  $p$  is the number of independent variables or features, and  $J$  is the cost function of the method (GÉRON, 2019). Equation (8) results in the variables going to zero but never there. This method deals better with problems that have the presence of multicollinear independent variables (GRUBER, 1998).

The Lasso regression, also known as L1 regularization, has the cost function given by

$$J = \text{MSE} + \lambda \sum_{j=1}^p |\beta_{1j}|. \quad (9)$$

The significant difference between Ridge and Lasso is that Lasso will shrinkage some features to zero (TIBSHIRANI, 1996). Because of that, Lasso is better em features selection for data sets with many features. The selection of the value  $\lambda$  is crucial to a good result. Using a  $\lambda$  too low will result in terms that do not correspond to the original data, and using a  $\lambda$  too high will cut some essential features. Because of that, some ML techniques help calibrate this parameter, tools, and techniques like split the data in training, test and validation data, cross-validation, and others are fundamentals for complex problems (GÉRON, 2019; BISHOP, 2006).

Since then, regression analysis has evolved a lot. Many methods emerged to solve different problems. Methods like the elastic net regression (HANS, 2011), principal components regression (LIU et al., 2003), support vector regression (SMOLA; SCHÖLKOPF, 2004), among others, have their theory and practical applications. However, in recent years, a method is getting attention, the sparse identification of nonlinear dynamics (SINDy). The original work of SINDy present the method and shows examples of inference nonlinear dynamics evolution law using data only.

## Dissertation Organization

This dissertation is organized such that, following this general introduction, the next chapter will comprehensively present the studied method. It will delve into the mathematical concepts involved, providing a didactic and profound understanding necessary for the method's application.

Following this, the benchmark systems employed to test SINDy are introduced. Four dynamic systems were chosen for this purpose. The primary and most utilized

in this work is the Duffing oscillator, selected for its nonlinear stiffness term and well-established physical applications. The next system is the simple pendulum, composed of only one trigonometric function, making it an ideal candidate to test the method's ability to identify such a function through polynomials, resembling a Taylor series. Lastly, two more dynamic systems, the Van der Pol oscillator, and the Rössler system, are presented. These two are used to demonstrate the method's versatility in inferring different dynamics and illustrate the shapes of attractors identified in chaotic cases.

Subsequently, the results of the mentioned tests for each benchmark are presented. Following this, a series of tests varying the quality of the data used were conducted on the Duffing oscillator. First, the result quality is tested by varying the number of data points used in the same time range. Second, by varying the intensity of the noise used in synthetic data. Lastly, by using the same amount of data points but captured at different time intervals. Each of these tests was also performed by varying the number of candidate mathematical functions to compose the inferred evolution law, aiming to determine how the presence of more functions would affect result quality.

Finally, conclusions from this work are presented, summarizing the main characteristics of the method, the use of different dynamic systems, and the results, with a focus on highlighting which data parameters most influenced the outcomes.

## 1 SPARSE IDENTIFICATION OF NONLINEAR DYNAMICS

The Sparse Identification of Nonlinear Dynamics (SINDy) method is a recent Machine Learning technique that employs regression fundamentals to discern the evolutionary laws of dynamical systems using only data, as described Brunton, Proctor and Kutz (2016). Since its inception, variations and hybrids of SINDy have been introduced by the original authors and others (BRUNTON; KUTZ; PROCTOR, 2017; ZHENG et al., 2019; BRUNTON et al., 2019; LYDON; POLAGYE; BRUNTON, 2023; JACOBS et al., 2023). Other authors, too, have articles and works using this method showing the technique’s potential (CORBETTA, 2020; FUKAMI et al., 2021; HONIGBAUM; ROCHINHA, 2022). The possibilities that arise with a new method, e.g., combining the ideas of different methods applying SINDy to verify the possibilities of improvements, like the sparse polynomial chaos expansion (ZENG et al., 2022) with SINDy, is an exciting idea.

However, different works address the inefficiency of SINDy in inferring the dynamic system from highly noisy data, irregular sampling frequencies, or missing values (YANG; MOHAMED; PERDIKARIS, 2020; MESSENGER; BORTZ, 2021; CORTIELLA; PARK; DOOSTAN, 2022; WENTZ; DOOSTAN, 2023). This results in unreliable inferred parameters or even the method’s inability to determine the correct dynamics, especially from real data. These studies present different methodologies applied alongside SINDy in an attempt to improve this deficiency, demonstrating how the method has much room for improvement.

This chapter will contemplate the fundamentals of the SINDy method and show others more recent possibilities with this algorithm.

Consider a one-dimensional autonomous dynamical system in which the state  $x(t)$  evolves according to the initial value problem

$$\dot{x}(t) = f(x(t)), \quad x(0) = a, \quad (10)$$

where the vector field (evolution law)  $f : \mathbb{R} \rightarrow \mathbb{R}$  is unknown.

Suppose the only known information about this system is a sample set with  $m$  measurements of  $x(t)$  and its temporal derivative  $\dot{x}(t)$  at instant  $t_1, t_2, \dots, t_m$ . The basic idea of SINDy is to reconstruct the original dynamics employing a regression, where the right-hand side of equation (10) is the approximation of a linear combination of certain elementary functions in a given user-defined library.

For instance, if this library of functions consists of polynomials up to degree  $n - 2$  and trigonometric functions such as sine and cosine of unit angular frequency, the following

regression problem must be solved

$$\begin{aligned}
\dot{x}(t_1) &\approx \xi_1 + \xi_2 x(t_1) + \dots + \xi_{n-2} x^{n-2}(t_1) + \xi_{n-1} \sin t_1 + \xi_n \cos t_1, \\
\dot{x}(t_2) &\approx \xi_1 + \xi_2 x(t_2) + \dots + \xi_{n-2} x^{n-2}(t_2) + \xi_{n-1} \sin t_2 + \xi_n \cos t_2, \\
\dot{x}(t_3) &\approx \xi_1 + \xi_2 x(t_3) + \dots + \xi_{n-2} x^{n-2}(t_3) + \xi_{n-1} \sin t_3 + \xi_n \cos t_3, \\
&\vdots \\
\dot{x}(t_m) &\approx \xi_1 + \xi_2 x(t_m) + \dots + \xi_{n-2} x^{n-2}(t_m) + \xi_{n-1} \sin t_m + \xi_n \cos t_m,
\end{aligned} \tag{11}$$

In matrix forms is given by

$$\begin{bmatrix} \dot{x}(t_1) \\ \dot{x}(t_2) \\ \dot{x}(t_3) \\ \vdots \\ \dot{x}(t_m) \end{bmatrix} \approx \begin{bmatrix} 1 & x(t_1) & \dots & x^{n-2}(t_1) & \sin t_1 & \cos t_1 \\ 1 & x(t_2) & \dots & x^{n-2}(t_2) & \sin t_2 & \cos t_2 \\ 1 & x(t_3) & \dots & x^{n-2}(t_3) & \sin t_3 & \cos t_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & x(t_m) & \dots & x^{n-2}(t_m) & \sin t_m & \cos t_m \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}, \tag{12}$$

or

$$\dot{\mathbf{x}} \approx \Theta(\mathbf{x})\Xi. \tag{13}$$

SINDy applies to first order dynamical systems of form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)), \tag{14}$$

where  $\mathbf{x}(t) \in \mathbb{R}^n$  is the state vector,  $d\mathbf{x}(t)/dt \in \mathbb{R}^n$  is the time-derivative of the state vector, and  $\mathbf{f}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the unknown evolution law.

A fundamental observation, in this case, is that higher-order systems usually, and surprisingly, have simplistic evolution laws, which a minimum set of elementary functions can represent. SINDy seeks to build this representation, promoting sparse solutions to the regression problem (BRUNTON; PROCTOR; KUTZ, 2016).

It is necessary to have measurements of the time series of the vectors  $\mathbf{x}(t)$  and  $\dot{\mathbf{x}}(t)$  to determine the evolution law  $\mathbf{f}$ . If the  $\dot{\mathbf{x}}(t)$  is not available by any factor, numerical differentiation is an alternative to obtain this data. These data need to be collected into matrix structures as follows,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_n(t_m) \end{bmatrix}, \tag{15}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \dots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \dots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \dots & \dot{x}_n(t_m) \end{bmatrix}. \quad (16)$$

After that, it is necessary to construct a library of candidate functions, denoted by  $\Theta(\mathbf{x})$ . Choosing polynomial, trigonometric, exponential functions, this library read as follows

$$\Theta(\mathbf{x}) = \begin{bmatrix} | & | & | & | & | & & | & | & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{\mathbf{P}_2} & \mathbf{X}^{\mathbf{P}_3} & \mathbf{X}^{\mathbf{P}_4} & \dots & \mathbf{X}^{\mathbf{P}_k} & \sin(\mathbf{X}) & \cos(\mathbf{X}) & e^{\mathbf{X}} \\ | & | & | & | & | & & | & | & | & | \end{bmatrix}. \quad (17)$$

Where each column in this matrix represents a candidate function, and the  $\mathbf{X}^{\mathbf{P}_n}$  notation indicates all possible  $n$ -order polynomials formed by combining the variables.

Generalizing the regression for the one-dimensional example, defined by equation (13), to the  $n$ -dimensional dynamical system of interest, the least-squares problem becomes

$$\dot{\mathbf{X}} \approx \Theta(\mathbf{X})\Xi, \quad (18)$$

where  $\Xi$  is a set of coefficients vectors as follows

$$\Xi = \begin{bmatrix} | & | & | & | \\ \Xi_1 & \Xi_2 & \dots & \Xi_n \\ | & | & | & | \end{bmatrix}, \quad (19)$$

where each column activates the candidate's functions for each column of  $\dot{\mathbf{X}}$ .

In more formal terms, this least-squares problem states

$$\Xi^* = \underset{\Xi}{\operatorname{argmin}} \left\| \dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi \right\|_2, \quad (20)$$

this equation aims to minimize the error between the observed derivatives  $\dot{\mathbf{X}}$  and the product of the library of functions  $\Theta(\mathbf{X})$  with the coefficients  $\Xi$ . This equation can be seen as a manifestation of the least squares optimization problem, which can be written more formally as:

$$\Xi^* = \underset{\Xi}{\operatorname{argmin}} \left\| \mathbf{E} \right\|_2^2, \quad (21)$$

where  $\mathbf{E} = \dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi$  is the residual error, and  $\|\cdot\|_2^2$  is the squared L2 norm.

The objective of this optimization problem is to find the coefficient vector  $\Xi$  that minimizes the squared residuals, hence producing the best fit between the observed and approximated dynamics.

Breaking down the error term  $\mathbf{E}$ :

$$\mathbf{E} = \dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi, \quad (22)$$

$$\mathbf{E}^T \mathbf{E} = (\dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi)^T (\dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi), \quad (23)$$

$$= \dot{\mathbf{X}}^T \dot{\mathbf{X}} - \Xi^T \Theta(\mathbf{X})^T \dot{\mathbf{X}} - \dot{\mathbf{X}}^T \Theta(\mathbf{X})\Xi + \Xi^T \Theta(\mathbf{X})^T \Theta(\mathbf{X})\Xi. \quad (24)$$

To find the optimal  $\Xi^*$ , you would differentiate this expression with respect to  $\Xi$  and equate it to zero, then solve for  $\Xi$ . This differentiation leads you back to the simplified equation provided:

$$\Xi^* = \underset{\Xi}{\operatorname{argmin}} \left\| \dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi \right\|_2, \quad (25)$$

indicating the optimal coefficients that result in the least squares fit between the observed and approximated dynamics (HASTIE; TIBSHIRANI; FRIEDMAN, 2009; NOCEDAL; WRIGHT, 2006).

To produce a sparse solution in machine learning, it is common to use a statistics technique known as Penalized regression, it is used where there are many predictors or features when some form of regularization is needed to prevent overfitting. This process aims to find the best-fitting model that explains the relationship between the independent variables and the dependent variable while penalizing overly complex models. This is achieved by adding a penalty term  $\lambda$  to the standard regression objective function, typically the sum of squared errors or the likelihood function.

To induce sparsity in machine learning solutions, practitioners commonly employ a statistical technique called penalized regression. This approach is utilized in scenarios with numerous predictors or features, requiring regularization to mitigate overfitting. Its objective is to discover the optimal model that elucidates the relationship between independent variables and the dependent variable while penalizing excessive model complexity. This process entails augmenting the standard regression objective function, typically the sum of squared errors or the likelihood function, with a penalty term denoted as  $\lambda$ .

In penalized regression with convex relaxation, the penalty term assumes the form of a convex function, facilitating sparsity in the model coefficients. The optimization objective can be expressed as

$$\Xi^* = \underset{\Xi}{\operatorname{argmin}} \left\| \dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi \right\|_2 + \lambda \operatorname{Penalty}(\Xi), \quad (26)$$

Here,  $\text{Penalty}(\Xi)$  represents the convex regularization term applied to the coefficients. Notable examples include LASSO (TIBSHIRANI, 1996), Least Angle Regression (LARS) (EFRON et al., 2004), and Basis Pursuit Denoising (BPDN) (CHEN; DONOHO; SAUNDERS, 2001). In the SINDy paper, Brunton, Proctor and Kutz (2016) introduced a novel methodology for promoting sparsity known as Sequential Thresholded Least Squares (STLS).

The STLS methodology consists of an iterative method that eliminates  $\Xi$ 's coefficients that are smaller than a threshold value  $\lambda$ . First is made a regression to obtain a non-sparse matrix  $\Xi$ . This matrix will have several spurious terms activating functions that do not compose the evolution law of the dynamics. Then, the module of each element of  $\Xi$  is thresholded with the value of  $\lambda$ , values of  $|\xi|$  that are smaller than  $\lambda$  will cut off, and the subsequent regression will not contain this mathematical function. This process continues until the convergence of  $\Xi$ .

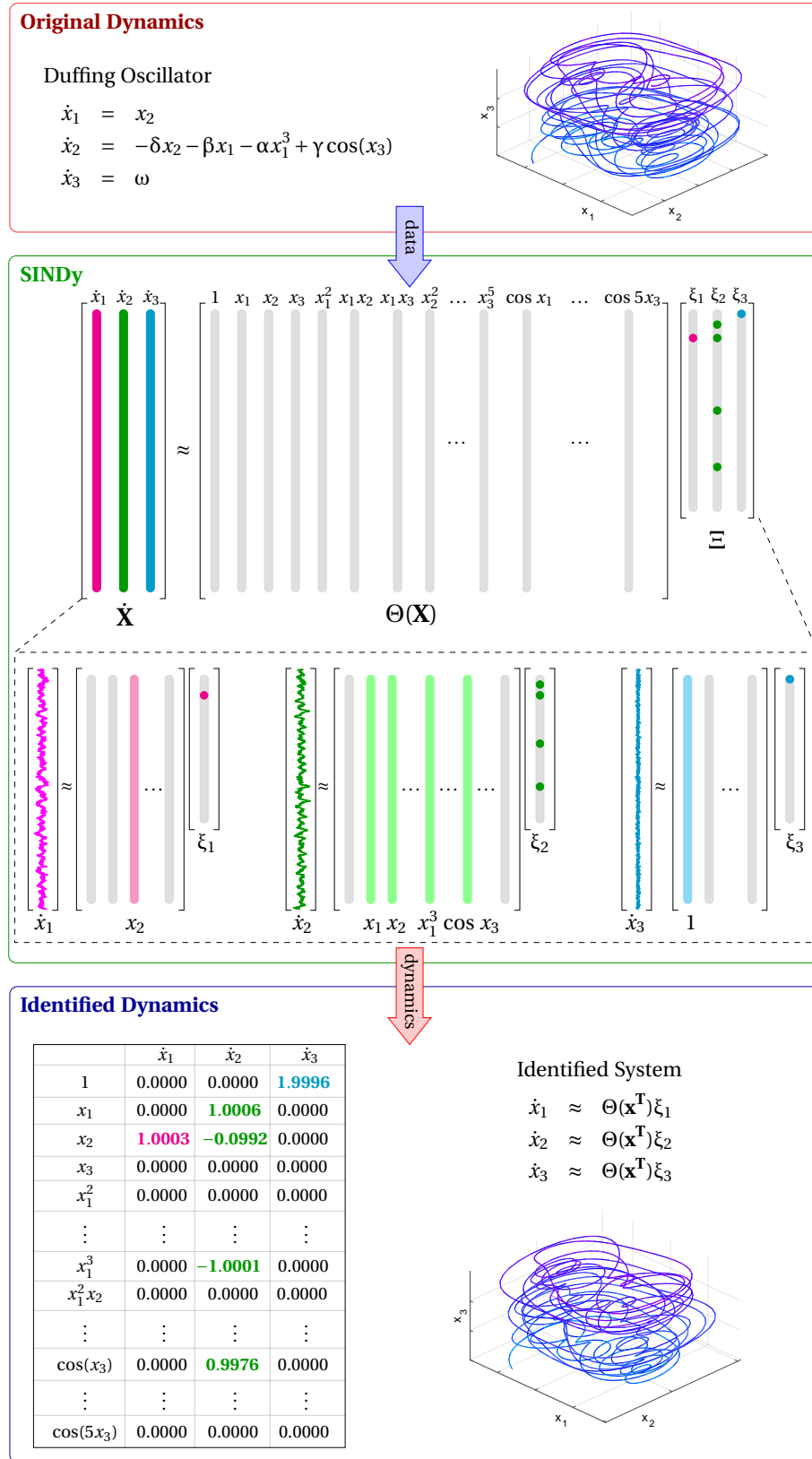
Knowing the magnitude of the dynamics is very important to calibrate the  $\lambda$  value. How the parameter  $\lambda$  acts as a direct threshold, parameters with the same or higher order of magnitude as the dynamics will eliminate original mathematical functions of the evolution law. When this information is missing, it is vital to use machine learning techniques (like cross-validation) to test different values and validate the lambda values chosen.

The difference between STLS and other classical promote sparsity methods like LASSO is that STLS is cheaper computationally and more precise for the dynamical systems tested in this works. Figure 2 exemplifies how the SINDy algorithm works to identify a Duffing oscillator for which measurements are available. The STLS in this example was able to infer the correct mathematical functions with the coefficients very close to the original.

The upper part of figure 2 is the time series of the dynamic following some evolution law. In that example is the data of a Duffing oscillator with chaotic behavior. That data is organized in a matrix as the equations (15) and (16) and given as input to the SINDy. Next, the user has to decide the mathematical functions composing the matrix of candidate functions. In that case, polynomials function until the fifth-order and cosine trigonometric functions compose the matrix. The data provided in a matrix and the sparsity promoter STLS (With a calibrated  $\lambda$ ) and the candidate functions, combining all that, the inferred dynamical system has the same mathematical functions as the original system that originated the data.



Figure 2 - SINDy schematics



Caption: Schematics of how the SINDy method works for the case of a Duffing oscillator.

## 2 BENCHMARKS SYSTEMS

As this work aims to verify the consistency and obtain insights into the SINDy method, it is crucial to use different dynamical systems to confirm this. Figure 3 shows the four dynamical systems present in this chapter. This chapter presents a brief introduction to all the dynamic systems used.

### 2.1 Duffing oscillator

The dynamical system chosen for most of the tests is the Duffing oscillator. This oscillator has many possible applications, such as structural dynamics (ZHANG et al., 2020), energy harvesting (COTTONE; VOCCA; GAMMAITONI, 2009; LOPES; PETERSON; Cunha Jr, 2017; PETERSON; LOPES; Cunha Jr, 2016; GUYOMAR et al., 2009; ROCA et al., 2019; NORENBORG et al., 2023), and complex and well-known dynamic behavior.

Figure 3a shows a schematic of a vibratory system that behaves like a Duffing oscillator. The dynamic behavior evolves according to

$$\ddot{x} + \delta \dot{x} + \alpha x + \beta x^3 = \gamma \cos(\omega t), \quad (27)$$

where the function  $x = x(t)$  is the displacement of the beam tip, and the  $\dot{x}$  and  $\ddot{x}$  are, respectively, the first and second derivative of  $x$ , i.e., the velocity and acceleration (BRENNAN; KOVACIC, 2011). Equation 27 can change to the first-order system given by

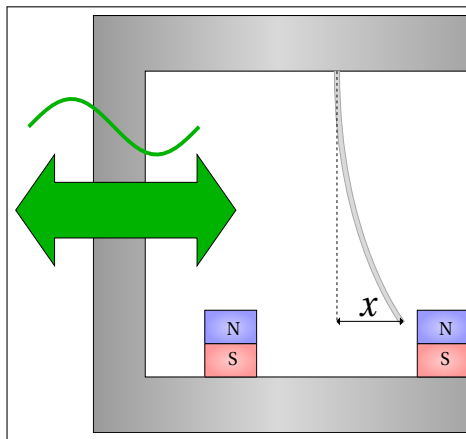
$$\begin{aligned} \dot{x}_1 &= \omega, \\ \dot{x}_2 &= x_3, \\ \dot{x}_3 &= -\delta x_3 - \alpha x_2 - \beta x_2^3 + \gamma \cos(x_1). \end{aligned} \quad (28)$$

The parameters in the equation of motion are:

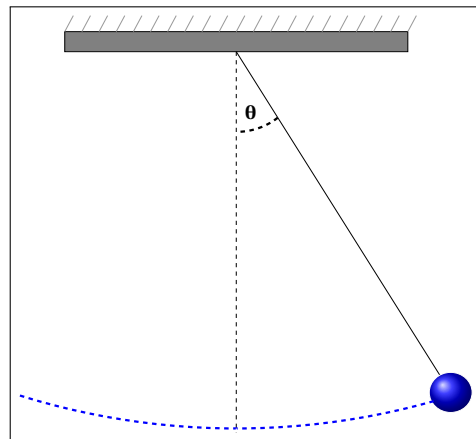
- The damping coefficient  $\delta$ .
- The linear stiffness  $\alpha$ .
- The nonlinear stiffness  $\beta$ .
- The external excitation amplitude  $\gamma$ .
- The external excitation frequency  $\omega$ .

Figure 3 - Benchmarks systems

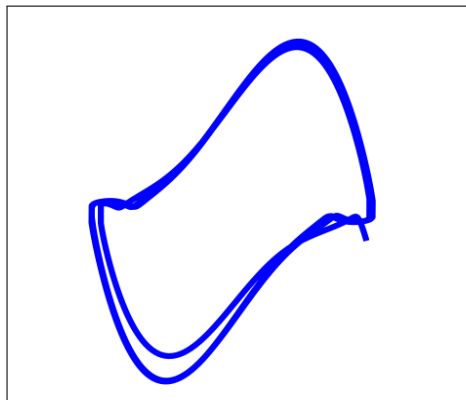
(a) Duffing oscillator schematic



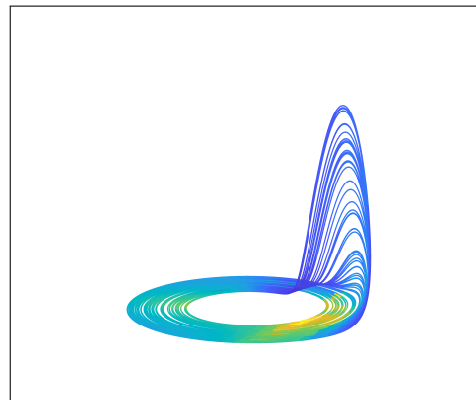
(b) Simple pendulum schematic



(c) Van der Pol dynamics



(d) Rössler dynamics



## 2.2 Simple pendulum

This system is the most basic trigonometric dynamical system. The equation is given by

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0, \quad (29)$$

where  $g$  is the acceleration of gravity,  $l$  is the length of the cord or rod, and  $\theta$  is the vertical angle between the actual position of the pendulum and the stationary point of equilibrium. Considering the  $l$  variable with the same length as the acceleration of gravity  $g$ , it is possible to simplify the evolution law to be dependent on only the  $\theta$  variable, resulting in

$$\frac{d^2\theta}{dt^2} = -\sin \theta, \quad (30)$$

or

$$\ddot{\theta} = -\sin \theta. \quad (31)$$

Figure 3b shows the schematics of that pendulum. Equation 31 can change to the first-order system given by

$$\begin{aligned} \dot{\theta}_1 &= \theta_2, \\ \dot{\theta}_2 &= -\sin \theta_1. \end{aligned} \quad (32)$$

## 2.3 Van der Pol Oscillator

The Van der Pol oscillator is another valuable benchmark system explored due to its rich and varied dynamical behaviors, making it a helpful model for the SINDy method. It has historical significance in nonlinear dynamics and finds applications in electronic circuits, cardiac dynamics, and other areas (POL, 1926; KENNEDY; CHUA, 1986; ZDUNIAK; BODNAR; FORYŚ, 2014).

Figure 3c a simplified representation of the Van der Pol dynamics. The governing equation for the time evolution of the system's state is given by

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0, \quad (33)$$

where:

- $x = x(t)$  is the state variable representing the displacement from the equilibrium.

- $\mu$  is a nonlinearity parameter.
- $\frac{dx}{dt}$  and  $\frac{d^2x}{dt^2}$  are the first and second-time derivatives of  $x$ , respectively.

Converting Equation 33 into a system of first-order ordinary differential equations (ODEs):

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - x_1.\end{aligned}\tag{34}$$

This conversion facilitates the application and analysis using the SINDy method.

## 2.4 Rössler System

The Rössler system is a well-studied and canonical example of a chaotic dynamical system, offering a wealth of complex dynamic behaviors, including chaos, complex dynamics, and fractals, with applications in various fields such as physics, engineering, and biology (Rössler, 1976; BETANCOURT-MAR; ALARCÓN-MONTELONGO; NIETO-VILLAR, 2005).

Figure 3d depicts a dynamic representation of the Rössler system. The dynamic equations governing the Rössler system are given by:

$$\begin{aligned}\dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c),\end{aligned}\tag{35}$$

where:

- $x, y$ , and  $z$  are the state variables.
- $a, b$ , and  $c$  are system parameters influencing the behavior and characteristics of the dynamic responses.

The Rössler system's nonlinearity and complex dynamics make it a crucial and insightful case for applying and analyzing the SINDy method.

### 3 RESULTS AND DISCUSSION

This chapter will discuss the results obtained by applying the SINDy method to the systems, as mentioned earlier. Initially, we will describe the process of data generation for the tests. Following this, we will present the results derived from applying SINDy to the Duffing and Van der Pol oscillators and the Rössler system. Subsequently, we will explore the Simple Pendulum case, accompanied by a detailed study examining how various parameters influence the outcomes generated by the method. All the MATLAB codes used for this thesis are available at [https://github.com/DiegoMSL/dissertation\\_codes.git](https://github.com/DiegoMSL/dissertation_codes.git).

#### 3.1 Generated Data

To generate synthetic data, we utilized MATLAB's ODE45 for numerically integrating the evolution law, resulting in the time series of the dynamical system. The process of simulating experimental measurements to produce this synthetic data involves three main steps:

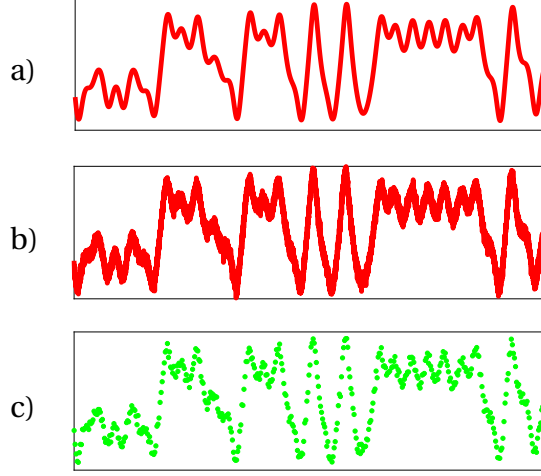
- Integrate the system dynamics using a small time step to achieve high-precision data.
- Introduce white Gaussian noise into the  $\dot{\mathbf{X}}$  data to simulate the fluctuations commonly found in experimental measurements. We achieve this by adding a matrix  $\mathbf{Z}$ , populated with zero-mean Gaussian entries, scaled by the noise intensity  $\sigma$ , to the time series.
- Selectively extract data points from the noise-infused time series to form a more sparse dataset suitable for the SINDy method.

Figure 4 illustrates the three stages involved in the synthetic data generation process.

#### 3.2 Analysis of the Duffing oscillator

We began by modeling a Duffing oscillator system, ensuring the selection of parameters that would preclude the system from descending into chaos. This culminated in the subsequent evolutionary rule:

Figure 4 - Generated data



Caption: The data generation process consists of three main steps. First, we utilize a numerical integrator to produce high-precision data. Next, we introduce Gaussian noise with an intensity of  $\sigma$  to mimic the data one might obtain experimentally. Finally, we curate a dataset with a more extended time interval than that used during the initial data generation.

$$\begin{aligned}
 \dot{x}_1 &= x_2, \\
 \dot{x}_2 &= -0.1x_2 + 1.0x_1 - 1.0x_1^3 + 1.0\cos(x_3), \\
 \dot{x}_3 &= 2.0.
 \end{aligned} \tag{36}$$

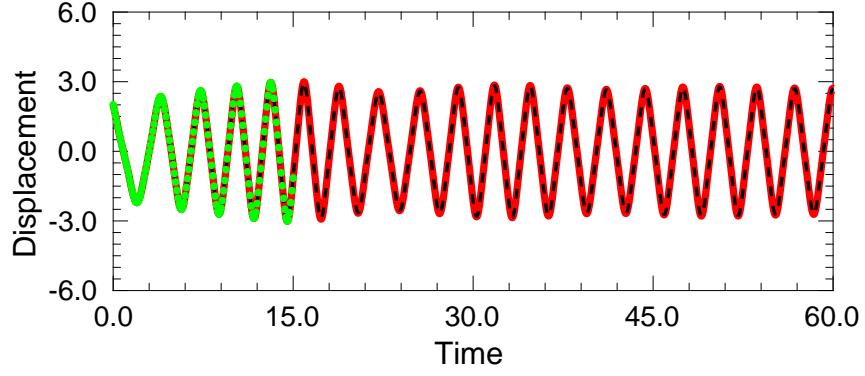
For the training phase, the initial conditions set for dimensions  $\dot{x}_1$ ,  $\dot{x}_2$ , and  $\dot{x}_3$  were 2.0,  $-2.0$ , and  $0.0$  respectively. We procured 151 samples, uniformly distributed from time 0 to 15, and introduced Gaussian noise at an intensity of 0.01. In terms of SINDy parameters, we opted for a function library encompassing power functions up to the fifth degree and cosine trigonometric functions through the fifth order. We designated the value of  $\lambda$  as 0.02. Consequently, the dynamic system derived was:

$$\begin{aligned}
 \dot{x}_1 &= 1.0003x_2, \\
 \dot{x}_2 &= -0.0992x_2 + 1.0006x_1 - 1.0001x_2^3 + 0.9976\cos(x_1), \\
 \dot{x}_3 &= 1.9996.
 \end{aligned} \tag{37}$$

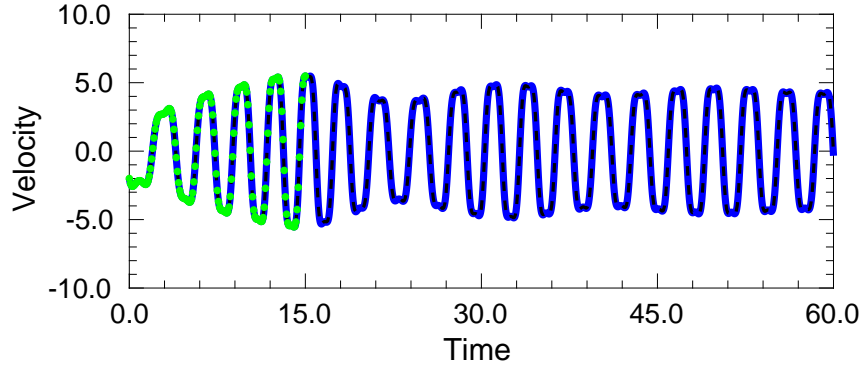
From our observations, the terms discerned through the method align with the foundational data, underscoring the precision and efficacy of the SINDy in this scenario.

Figure 5 - Duffing oscillator numerical dynamics compared with data driven dynamics

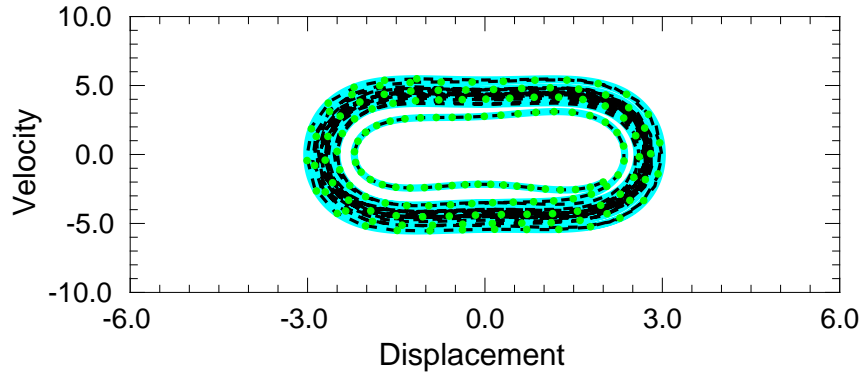
(a) Displacement time series



(b) Velocity time series



(c) Phase space

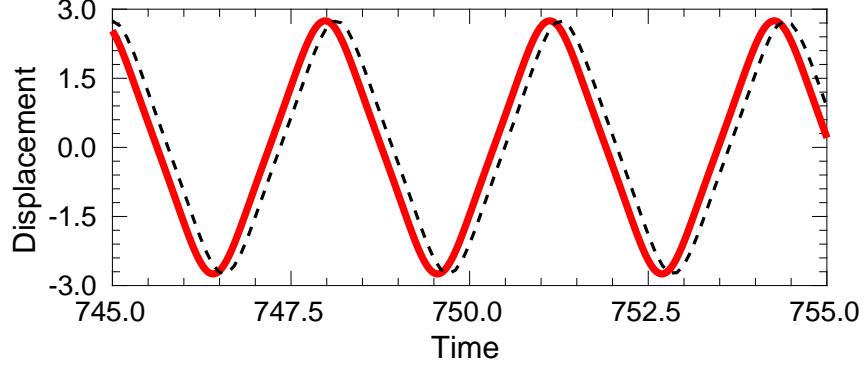


Caption: The figures display the displacement, velocity, and phase space of the non-chaotic Duffing oscillator. We used 151 data points, depicted as green dots, for the method. The continuous colored lines illustrate the numerical system, while the black dashed lines correspond to the results inferred by the SINDy method.

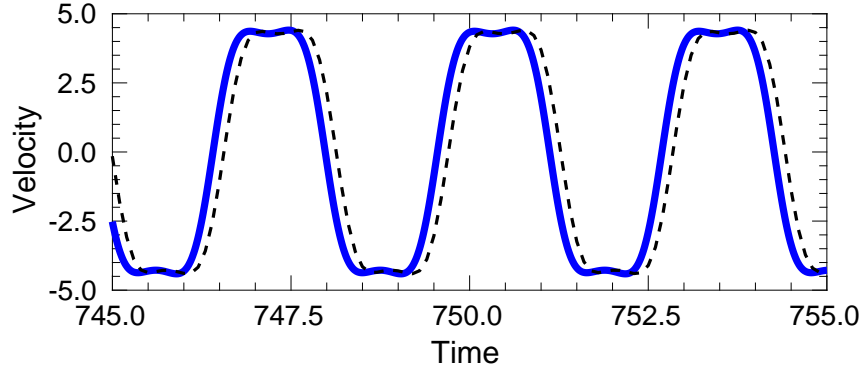


Figure 6 - Differences in numerical dynamics versus data-driven approaches over extended durations

(a) Displacement time series



(b) Velocity time series



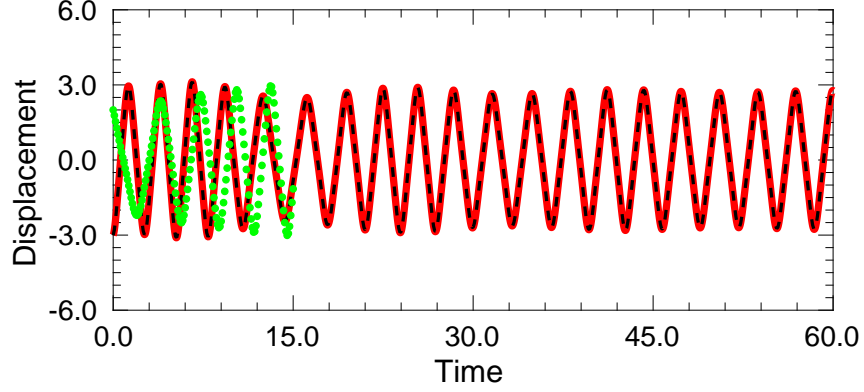
Caption: An excerpt from the Duffing oscillator's time series emphasizes that, over prolonged periods, the results from SINDy reveal notable discrepancies. These differences lead to a misplacement between the two dynamics.

Visual representation of this data can be gleaned from figures 8a, 8b and 8c. Here, the original system dynamics are depicted as a continuous colored line, the SINDy-identified system as a dashed line, and the set of 151 data points are marked in green. While the dynamic representations largely coincide up to time 60, disparities in the coefficients predictably lead to a phase drift in the identified system over an extended period, as elucidated in figures 6a and 6b.

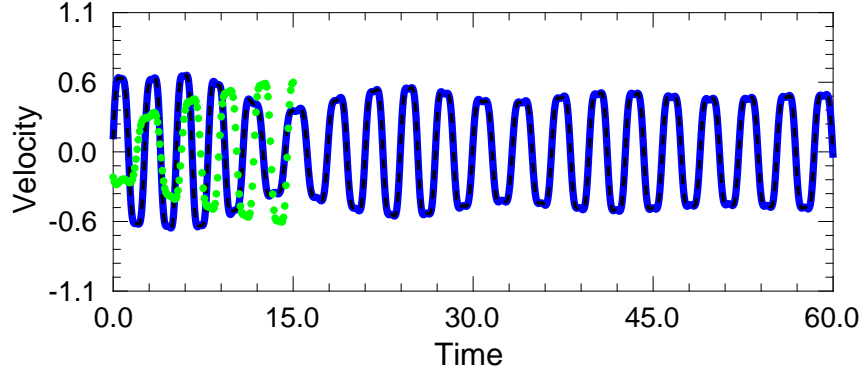
Another crucial aspect of the method emerges in the subsequent test. For this, we again employ the parameters and initial conditions from the non-chaotic Duffing test to train the method. We then set the initial conditions  $\dot{x}_1$ ,  $\dot{x}_2$ , and  $\dot{x}_3$  to values of  $-3.0$ ,  $1.0$ , and  $0.0$  respectively. This adjustment allows us to validate the model and compare it to the numerical results. In Figure 7, the green dots correspond to the 151 data samples from the method's initial training condition. The represented lines showcase the dynamics under the new initial condition: the continuous line for the numerical dynamics and the

Figure 7 - Duffing Oscillator Numerical Dynamics Compared with Data-Driven Training with Different Initial Conditions

(a) Displacement Time Series



(b) Velocity Time Series



Caption: The figures illustrate the displacement and velocity of the non-chaotic Duffing oscillator. 151 data samples from the same oscillator but with different initial conditions were used, as indicated by the green points exhibiting distinct behavior from the identified and numerical dynamics.

dashed line for the inferred method's result. This clear delineation indicates that the SINDy result, derived from this initial condition, holds validity even for alternative initial conditions. Given that the SINDy method determines the parameters of the evolutionary law, a successful inference process ensures that alterations in initial conditions will not impact the training outcome.

The Duffing oscillator exhibits a wide range of dynamic behaviors. By adjusting certain parameters and setting specific initial conditions, the system can achieve chaotic dynamics. The following equations describe this behavior:

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -0.1x_2 + 1.0x_1 - 0.25x_1^3 + 2.5 \cos(x_3), \\
\dot{x}_3 &= 2.0.
\end{aligned} \tag{38}$$

The system starts with conditions for  $x_1$ ,  $x_2$ , and  $x_3$  at 1.0,  $-1.0$ , and 0.0 respectively. Using 151 samples spanning 0 and 15, we incorporated a Gaussian noise of magnitude 0.01 and set a sparsity parameter,  $\lambda$ , to 0.02. Our functional library included polynomial functions up to the fifth order and trigonometric cosine functions, which led to the production of the following dynamic system:

$$\begin{aligned}
\dot{x}_1 &= 1.0003x_2, \\
\dot{x}_2 &= -0.0992x_2 + 0.9995x_1 - 0.2501x_2^3 + 2.4999 \cos(x_1), \\
\dot{x}_3 &= 1.9996.
\end{aligned} \tag{39}$$

Even though the system behaves chaotically, SINDy's inferences align closely with it, displaying parameters that resemble those from the numerical data. As shown in the Figure 8, minor variations in these parameters can result in significant disparities, significantly beyond the range of the training data in both temporal and phase spaces, a hallmark of chaotic systems.

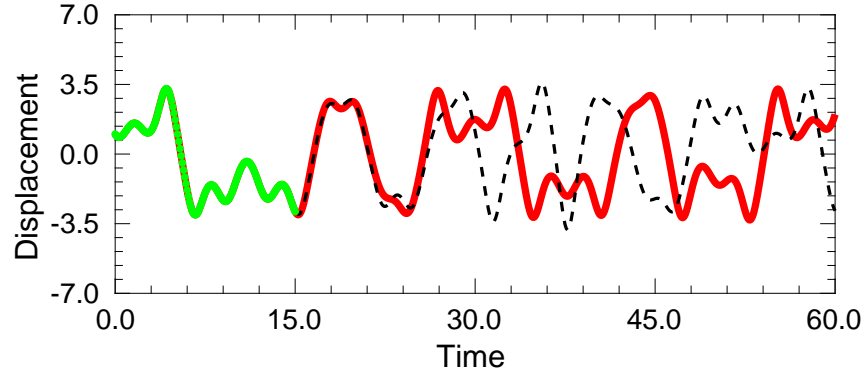
The Duffing oscillator offers many test possibilities, and the ones mentioned are just a few among many. For a deeper dive into various configurations, one can refer to the works by Lopes and Cunha Jr (2019) and Cunha Jr and Lopes (2021). In a distinct study, Lopes and Cunha Jr (2022) scrutinized the physical consistency of sparse regressions with SINDy, focusing on the disparities in energy and momentum balance of the detected system compared to its original — both in contexts with pristine data and those tainted by noise. Notably, as time progressed, these differences became more pronounced. This amplification was particularly evident when Gaussian noise was present, revealing a proportional relationship between the noise intensity and the growing discrepancies.

### 3.3 Analysis of the pendulum dynamics

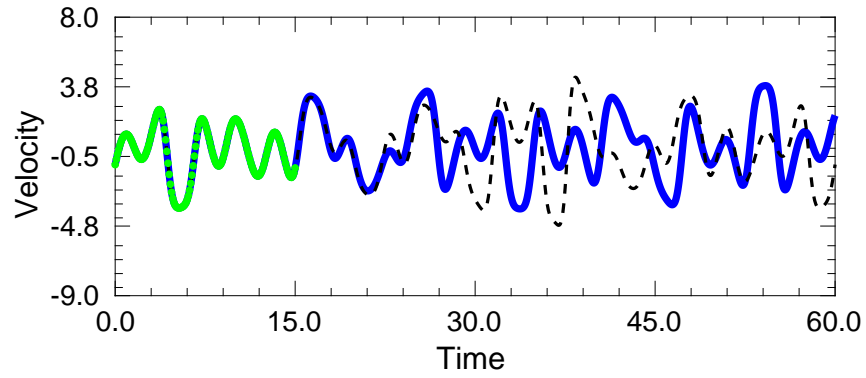
Another interesting test involves verifying if the SINDy method can capture the behavior of trigonometric functions using only polynomial functions in the function library,

Figure 8 - Comparison between the chaotic Duffing oscillator's numerical dynamics and its data-driven dynamics

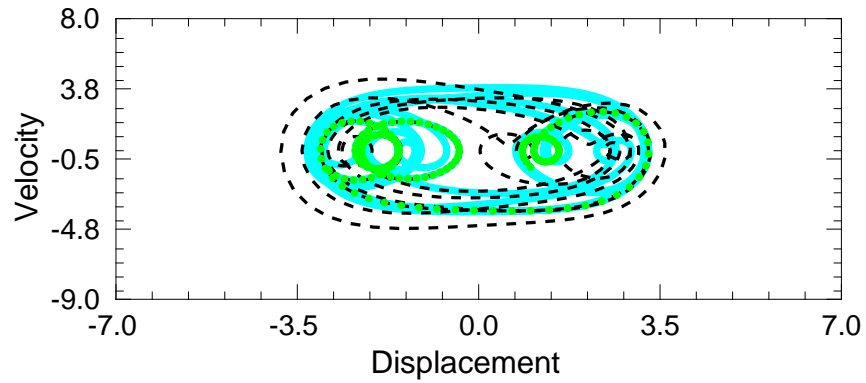
(a) Displacement time series



(b) Velocity time series



(c) Phase space



Caption: These visualizations capture the displacement, velocity, and phase space dynamics of the chaotic Duffing oscillator. We used 151 data points, represented as green dots, for the method. Continuous colored lines represent the numerical system, while black dashed lines outline the results inferred using the SINDy method.

approximating them through the Taylor series. As a result, Equation 32 transforms into:

$$\begin{aligned}\dot{\theta}_1 &= \theta_2, \\ \dot{\theta}_2 &= -\theta_2 + \frac{\theta_2^3}{6} - \frac{\theta_2^5}{120} + O(\theta_2^7),\end{aligned}\tag{40}$$

or

$$\begin{aligned}\dot{\theta}_1 &= \theta_2, \\ \dot{\theta}_2 &= -\theta_2 + 0.1667\theta_2^3 - 0.0083\theta_2^5 + O(\theta_2^7).\end{aligned}\tag{41}$$

We used ODE45 to select 151 equally spaced samples over the time interval from 0 to 15, with initial conditions for  $\dot{\theta}_1$  and  $\dot{\theta}_2$  set to 1.0 and 1.0, respectively. We deliberately avoided introducing Gaussian noise into the data to ensure maximum data precision and to verify accuracy. Polynomial functions up to the fifth order were employed to compare them with the inferred parameters to those of Equation 41. Using a lambda ( $\lambda$ ) value of 0.001, the method's result is:

$$\begin{aligned}\dot{\theta}_1 &= 1.0000\theta_2, \\ \dot{\theta}_2 &= -0.8599\theta_2 + 0.0937\theta_2^3 - 0.0730\theta_1\theta_2^2 - 0.0083\theta_2^5.\end{aligned}\tag{42}$$

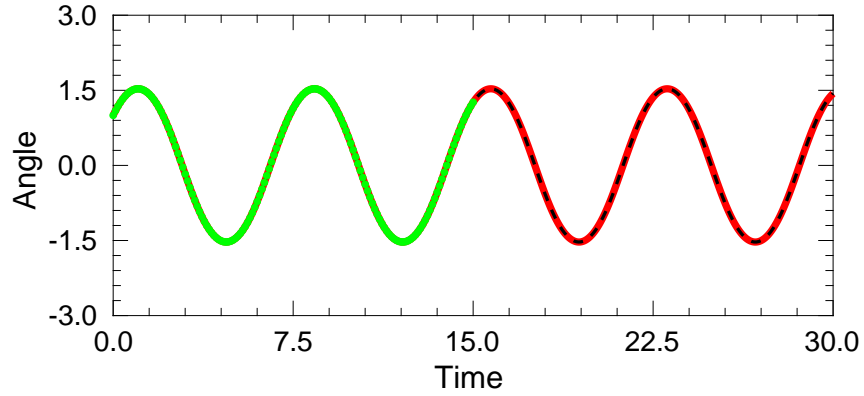
Notice that the method identified a dissipative polynomial that should not be present, specifically, the  $-0.0730\theta_1\theta_2^2$  term. Since the numerical ODE45 method is dissipative, we speculated that SINDy might have identified this additional term. To test this hypothesis, we used the ODE78 method (EAGLE, 2023), known for having less dissipation, to see if it would alter the parameters or even exclude the term. However, the result remained precisely the same. We also tested variations by altering the number of collected data points and the data collection time interval, but none of these scenarios altered the inferred evolution law. Figure 9 demonstrates that, despite the dissipative term, the inferred dynamics correspond well to the original dynamics within a short interval.

### 3.4 Analysis of the Van der Pol oscillator

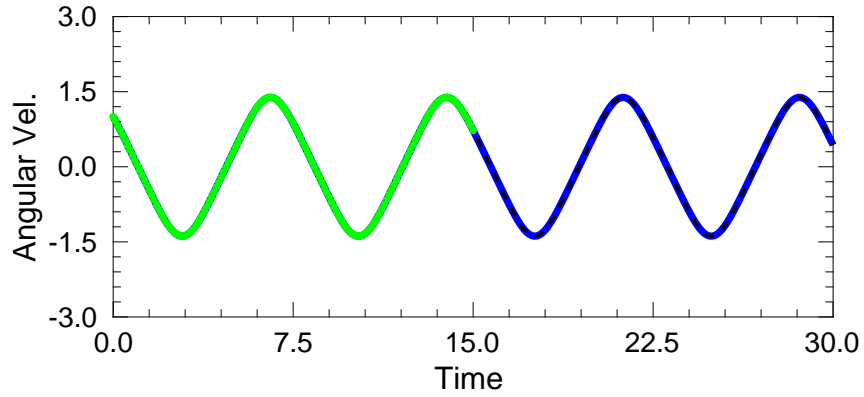
Considering the Van der Pol oscillator system shown in the previous chapter, we used Equation 34 to generate 151 samples between 0 and 15. The library of polynomial functions consists of polynomials up to the fifth order without trigonometric terms. The initial condition is given as  $x_1 = 0.5$  and  $x_2 = -0.2$ . Gaussian noise with an intensity of 0.1 and an STLS dispersion parameter of 0.5 were used. The result obtained for the

Figure 9 - Time series of the simple pendulum with a Taylor series

(a) Angular time series

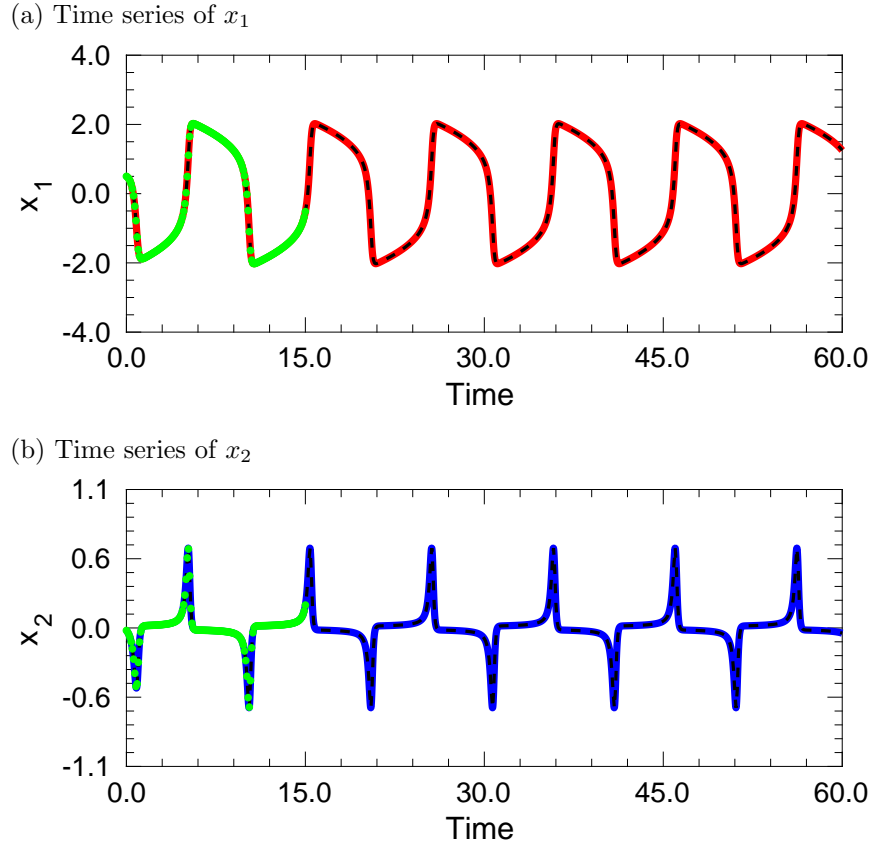


(b) Angular Velocity time series



Caption: Comparison of numerical time series with data-inferred ones, where the numerical series uses the sine trigonometric function, and the inferred series uses Taylor polynomials up to the fifth order for the sine function. Represented by the green points, a total of 151 data samples were used.

Figure 10 - Van der Pol oscillator numerical dynamics compared with data-driven dynamics



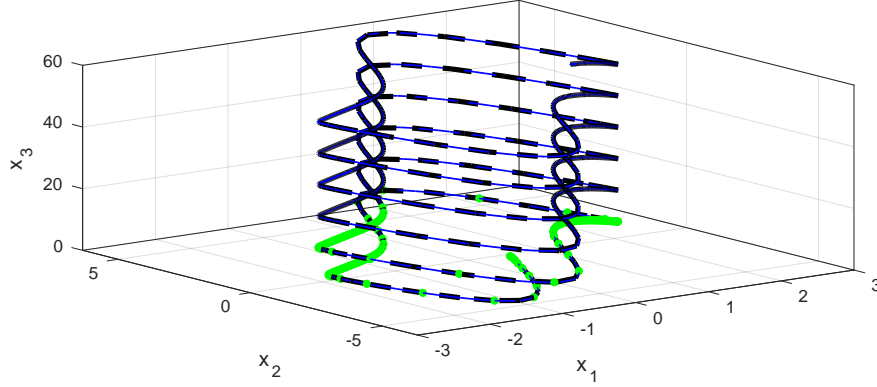
Caption: Time series of dimensions  $x_1$  and  $x_2$  of the Van der Pol system. The colored solid line represents the original dynamics that generated the data, the 151 green points are the samples used in the SINDy method, and the dashed line is the dynamics identified by the method.

system with  $\mu = 4$  was

$$\begin{aligned} \dot{x}_1 &= 0.9997x_2, \\ \dot{x}_2 &= -1.0000x_1 + 4.0002x_2 - 4.0005x_1^2x_2. \end{aligned} \tag{43}$$

Similar to the Duffing oscillator, the Van der Pol system had its dynamics accurately inferred by the SINDy method. The results of the time series and phase space are shown in the following figures.

Figure 11 - Three-dimensional phase space of the Van der Pol oscillator



Caption: Phase space of the Van der Pol oscillator comparing the original trajectories with those identified by the data, represented by the green points.

### 3.5 Analysis of the Rössler system

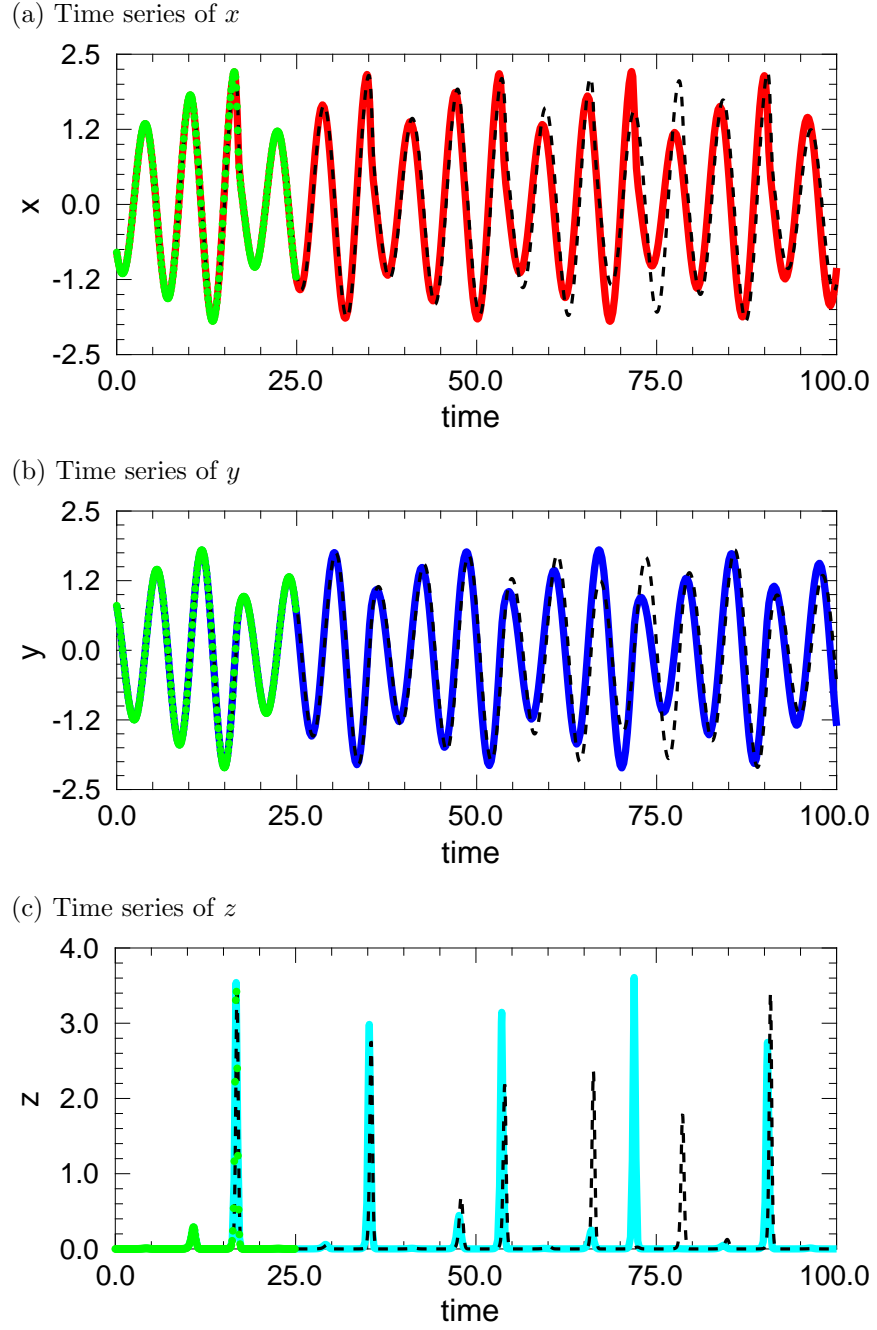
The choice of the Rössler System as the final system in the Benchmark Systems chapter is due to its dynamic behavior. We use Equation 35 with the following parameters:  $a = 0.1$ ,  $b = 0.1$ , and  $c = 14$ . The system exhibits chaos starting from the initial condition  $x_0 = -8$ ,  $y_0 = 8$ , and  $z_0 = 0$ . We use 250 samples for this simulation, equally distributed between 0.001 and 25, with Gaussian noise of intensity 0.5. We also construct a library of polynomial functions up to the fifth order with a dispersion parameter of  $\lambda = 0.05$ . The result is the following dynamics:

$$\begin{aligned}\dot{x} &= -0.9922y - 1.0063z, \\ \dot{y} &= 0.9954x + 0.0969y, \\ \dot{z} &= 0.0938 - 14.0639z + 1.0037xz.\end{aligned}\tag{44}$$

Once again, SINDy correctly identifies the dynamics. However, as this system is chaotic, the time series rapidly diverge. This divergence is illustrated in the figures 12, which compare the original dynamics with the identified dynamics of the three dimensions:  $x$ ,  $y$ , and  $z$  of the system. Despite these differences, when we examine the Rössler attractor, as shown in figure 13, the shape of the attractor reconstructed by the identified system closely resembles that of the original system. This demonstrates that despite variations in parameters and time series, SINDy effectively captures the behavior of the chaotic dynamics.

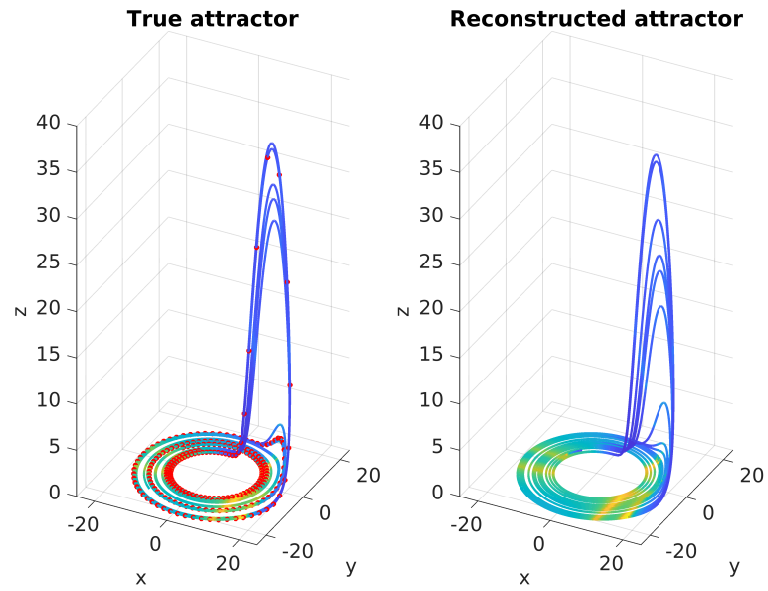


Figure 12 - Time Series of the Three Dimensions of the Rössler System



Caption: Comparison of the identified time series (black dashed line) with the numerical counterpart (colored solid line) along with the 250 samples (green dots) used to feed the SINDy method. Due to the chaotic nature of the system, it can be observed that after some time, the identified dynamics diverges from the numerical, especially in the  $z$  dimension.

Figure 13 - Rössler Attractor That Originated the Data Compared to the One Reconstructed by the Identified Dynamics



Caption: The left attractor (True attractor) illustrates the phase space of the original dynamics, where the red dots represent the 250 samples used by the SINDy method to infer the evolution law. On the right is the reconstructed attractor based on the inferred dynamics; it can be observed that despite the chaotic nature of the dynamics, the shapes of the attractors exhibit similar characteristics.

### 3.6 Convergence Test for Number of Simulations

In the following section, a series of statistical tests varying key parameters in SINDy simulations will be conducted, and the quality of the results will be assessed. First and foremost, it is crucial to establish the minimum number of simulations required to achieve satisfactory statistical outcomes. To address this, the average Root Mean Squared Error (RMSE) between the numerical dynamics and those inferred from the Duffing oscillator data is examined by iteratively applying the SINDy method 10, 25, 50, 100, 250, and 500 times.

The initial condition  $[1.0, -1.0, 0.0]$  for dimensions  $x_1$ ,  $x_2$ , and  $x_3$  is employed to generate training data based on Equation 38. The data is evenly distributed over 301 samples spanning from 0 to 30. This ample dataset is chosen to minimize the likelihood of inferring evolution laws different from those that generated the data, thus avoiding outliers. Despite the same initial condition and evolution law, each inference involves generating Gaussian noise, ensuring unique noise patterns for each run and resulting in slight fluctuations in the parameters of the inferred evolution law.

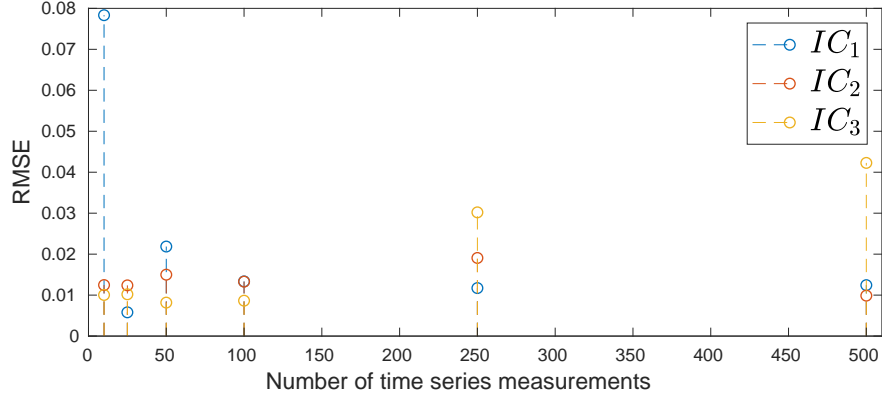
Subsequently, three additional initial conditions  $IC_1 = [2, 0, 0]$ ,  $IC_2 = [0, -2, 0]$ , and  $IC_3 = [-2, 0, 0]$  are used in the dynamics inferred by SINDy and compared against their numerical counterparts. For this evaluation, 1001 samples between 0 and 100 are utilized, and the average RMSE for each data point across the entire dynamics is computed. This process is repeated for varying numbers of repetitions, with the final step involving the calculation of the average of all RMSE means. The outcomes of this test are presented in Figure 14, illustrating these averages for each initial condition across different simulation quantities. Notably, only  $IC_1$  for 10 repetitions displayed elevated RMSE, indicating sensitivity to potential outlier results with only 10 simulations. Despite expectations that increasing the number of simulations would yield improved results beyond a certain threshold, this was not consistently observed across all initial conditions. This suggests that a higher computational workload may not necessarily translate to enhanced results. As satisfactory outcomes were observed with 25 to 100 simulations, 100 simulations were chosen as the standard to ensure a robust statistical evaluation without excessive computational burden.

### 3.7 Analysis of the RMSE and Correlation

These tests use a variety of parameters numbers, such as:

- Number of data points,
- noise intensity,

Figure 14 - Average RMSE for Different Numbers of Time Series



Caption: Results of three dynamics with different initial conditions, depicting the average RMSE between the identified and numerical dynamics for six simulation quantities (10, 25, 50, 100, 250, and 500).

- the time interval of the data.

To calculate the influence of parameters in the result, the root-mean-squared error (RMSE) and correlation between the original and identified dynamics. The RMSE is a statistical tool to measure the distance of a model with the data given by one equation. This equation is important not only for statistical but for machine learning. Some techniques use the RMSE as a calibration tool, minimizing the RMSE, which normally results in better models. The correlation, or dependence, is also a statistical tool. This measure the relationship between two variables given by a complex equation. The result of this equation varies between  $-1$  and  $1$ , where closer to  $1$  is the correlation between more linear and noisiness, and closer to  $-1$  is the inverse linear and noisiness, if the result is zero, the variables do not correlate.

Therefore the Duffing oscillator is again used for these tests because of the complex dynamics and a well-known system. The parameters of the dynamical system are selected in a way that chaos is avoided because the nature of chaotic systems of any variation results in a divergence after some time. The system identified is given by

$$\begin{aligned}
 \dot{x}_1 &= x_2, \\
 \dot{x}_2 &= -0.1 x_2 + 1.0 x_1 - 1.0 x_1^3 + 2 \cos x_3, \\
 \dot{x}_3 &= 2.0,
 \end{aligned} \tag{45}$$

this equation and initial condition  $[1, -1, 0]$ . Three different initial conditions are used to validate the result. for each of the three variation parameters, the number of candidate functions is changing too.

Table 1 - RMSE for different number of data points with poly order 3

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.27006	0.30438	0.27924	0.24941	0.21911	0.17059	0.13369
$IC_2$	0.03620	0.03004	0.02483	0.01907	0.01795	0.01338	0.00878
$IC_3$	0.07226	0.05815	0.04680	0.03342	0.02478	0.01969	0.01410

Table 2 - RMSE for different number of data points with poly order 4

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.29546	0.31513	0.32203	0.29607	0.24659	0.18359	0.15996
$IC_2$	0.06064	0.04338	0.05791	0.03228	0.01795	0.03486	0.02954
$IC_3$	0.088	0.07123	0.10707	0.07728	0.07030	0.04865	0.10156

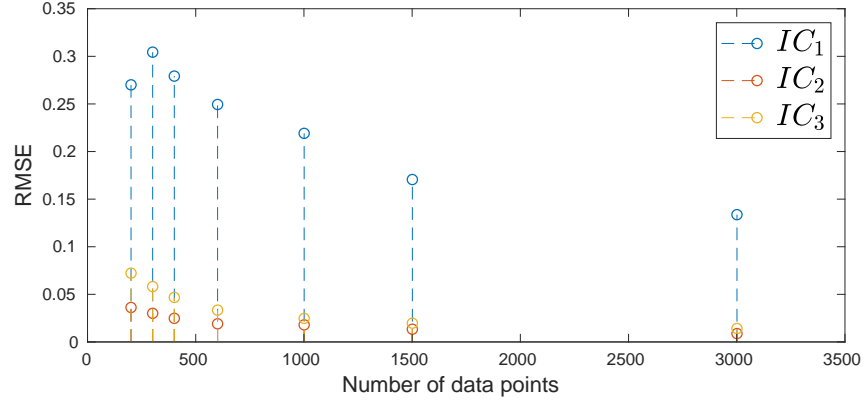
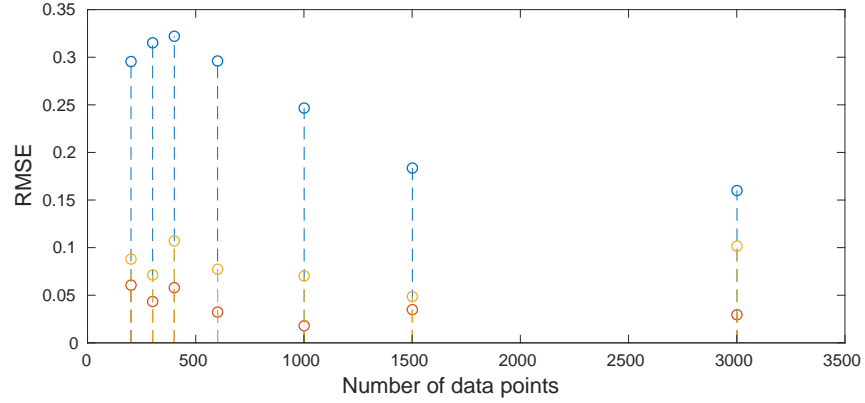
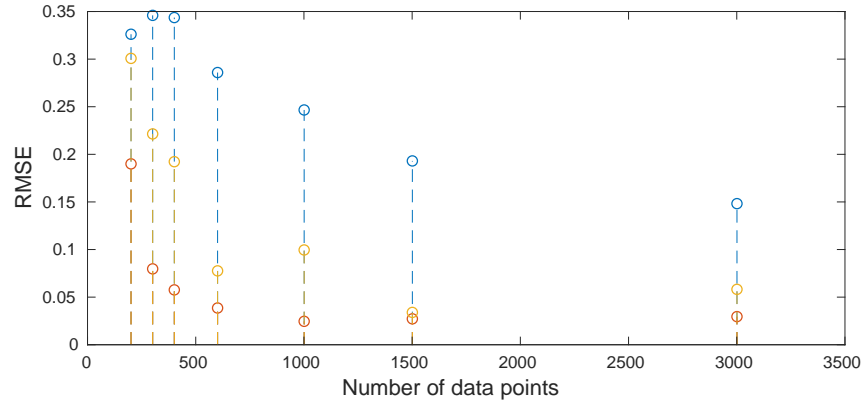
### 3.7.1 Number of data points

In this case, we examine the impact on result quality by varying solely the number of equally spaced data points obtained within the time interval of 0 to 30. Databases with 200, 300, 400, 600, 1000, 1500, and 3000 points were employed, all originating from the same initial condition dynamics  $[1, -1, 0]$ . Following the database generation, Gaussian noise with an intensity of 0.005 is introduced. A sparsity parameter of 0.085 is consistently applied across all simulations.

Once the dynamic system is identified, we select 1001 points within the time span from 0 to 100 in both the identified and numerical dynamics. However, these are determined using three distinct initial conditions:  $IC_1 = [1, 0, 0]$ ,  $IC_2 = [1.5, -0.5, 0]$ , and  $IC_3 = [-2, 1, 0]$ . These points are then compared, allowing us to compute the RMSE and correlation for these dynamics. The RMSE results are visually presented in Figures 15a, 15b, and 15c, with detailed tabular information available in Tables 1, 2, and 3. The distinction among these three results lies in the number of polynomial functions included in the library for determining the inferred dynamics. Figure 15a and Table 1 correspond to dynamics with a library comprised of combinations of polynomials up to the third order, totaling 20 potential polynomial functions excluding trigonometrics. Similarly, Figure 15b and Table 2 utilized polynomials up to the fourth order, while Figure 15c and Table 3 encompassed polynomials up to the fifth order.

Analyzing the results in Table 1, it is evident that, overall, the RMSE value decreases as the number of data points increases. This trend, however, does not hold true for the variation between 200 and 300 points for  $IC_1$ , where there is a 12.7% increase in RMSE. When comparing results from 200 to 3000 data points for different initial conditions, there are significant reductions of 50.50%, 75.73%, and 80.49%, respectively. Despite the improvement in RMSE with an increased number of data points, a straight-

Figure 15 - RMSE with different number of data points

(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: RMSE between the identified and original Duffing oscillator with different number of data points varying the number of candidate functions, each result is the mean of one hundred simulations for three different initial conditions.

Table 3 - RMSE for different number of data points with poly order 5

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.32613	0.34596	0.34366	0.2859	0.24659	0.19308	0.14821
$IC_2$	0.18997	0.07969	0.05761	0.03860	0.02459	0.02720	0.02954
$IC_3$	0.30077	0.22143	0.19227	0.07760	0.09954	0.03394	0.05818

Table 4 - Correlation  $\dot{x}_1$  for different number data points with poly order 3

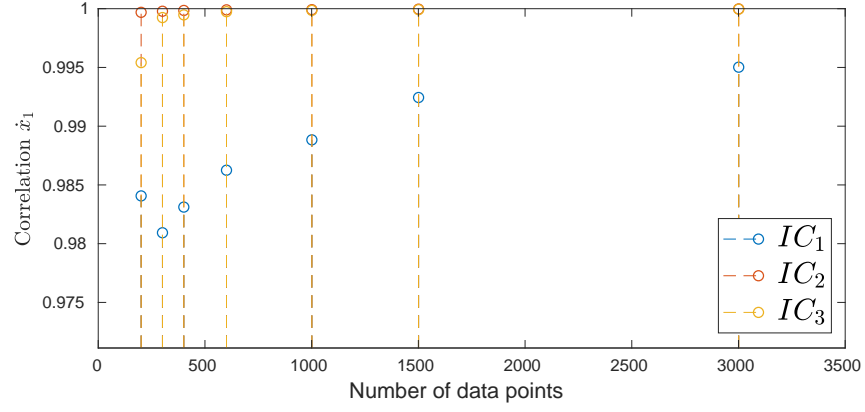
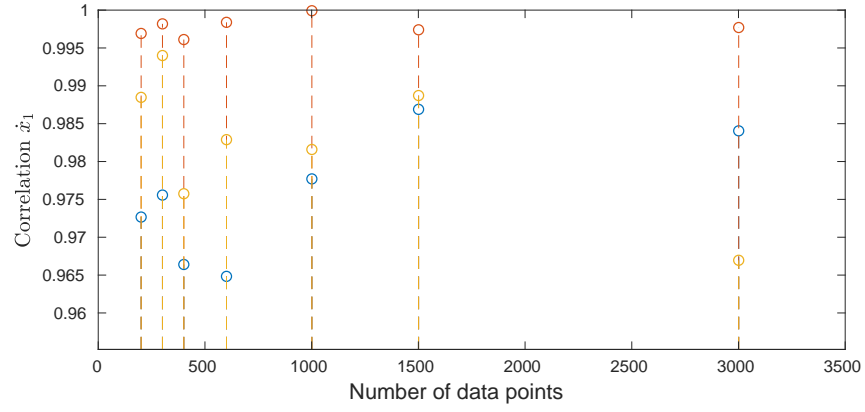
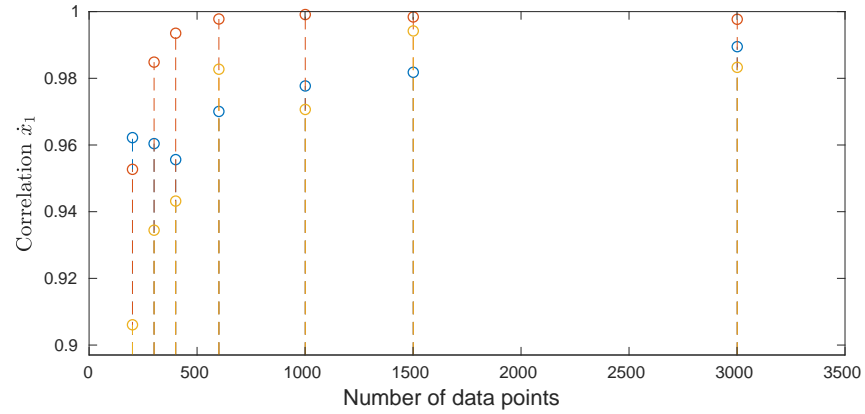
Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.98407	0.98093	0.98311	0.98625	0.98884	0.99244	0.99502
$IC_2$	0.99969	0.99978	0.99985	0.9999	0.99992	0.99996	0.99998
$IC_3$	0.99541	0.99924	0.99947	0.99975	0.99986	0.99992	0.99996

forward correlation between the number of data and RMSE value is not observed. For instance, examining the results for  $IC_2$ , an increase from 200 to 300 points yields a 50% rise in the number of points and a 17.03% reduction in RMSE. Similarly, between 300 and 400 points, a 33.33% increase results in a 17.34% RMSE reduction, and between 600 and 1000 points, with a 66.67% increase, there is only a  $-5.91\%$  RMSE decrease.

Turning to Figures 15b and 15c, along with Tables 2 and 3, most considerations made for the results with a library up to the third-degree polynomial remain valid. However, more increases in the RMSE value are observed when increasing the number of points. Nevertheless, when comparing between 200 and 3000 data points, a notable drop in RMSE is observed, except for  $IC_3$ , which experiences a 108.77% increase. This high value suggests that, in some simulations, SINDy failed to accurately determine the dynamics, resulting in an outlier. Disregarding this result and comparing 200 with 1500 data points, there is a  $-44.72\%$  drop in RMSE.

Next, examining the results with the difference in the number of functions in the library, a comparison between the results in Tables 1 and 2 reveals that most values increased, with only one remaining the same.  $IC_3$  with 3000 data points experienced the highest proportional increase, where 0.01410 RMSE increased to 0.10156, marking a 620.44% increase. Individually analyzing the values in search of any correlation reveals that, for example, for  $IC_2$ , an increase in functions yielded the same RMSE value. Comparing Tables 2 and 3, once again, most values increased with the addition of functions. However, there were cases where the result was better with more functions, such as the case with 3000 data points, which experienced a  $-42.71\%$  decrease. Instances where the results improved were less frequent, and the relative improvement in percentage was less pronounced, with the highest increase in RMSE being 241.78% between the values of 0.08800 and 0.30077 for  $IC_3$  with 200 data points.

The correlation analysis was divided into four distinct evaluations. The results

Figure 16 - Correlation of  $\dot{x}_1$  with different number of data points(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: Correlation of the  $\dot{x}_1$  variable between the identified and original Duffing oscillator with different number of data points varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.



Table 5 - Correlation  $\dot{x}_1$  for different number data points with poly order 4

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.97267	0.97557	0.96640	0.96483	0.97771	0.98689	0.98405
$IC_2$	0.99692	0.99818	0.99611	0.99839	0.99992	0.99740	0.99770
$IC_3$	0.98848	0.99400	0.97575	0.98289	0.98159	0.98872	0.96696

Table 6 - Correlation  $\dot{x}_1$  for different number data points with poly order 5

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.96221	0.96041	0.95562	0.97006	0.97771	0.98180	0.98949
$IC_2$	0.95270	0.98485	0.99351	0.99779	0.99912	0.99839	0.99770
$IC_3$	0.90608	0.93445	0.94318	0.98272	0.97065	0.99421	0.98328

Table 7 - Correlation  $\dot{x}_2$  for different number data points with poly order 3

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.95845	0.94997	0.95565	0.96374	0.97057	0.98009	0.98689
$IC_2$	0.99936	0.99955	0.99969	0.9998	0.99983	0.99992	0.99996
$IC_3$	0.99391	0.99844	0.99891	0.99948	0.99970	0.99983	0.99991

Table 8 - Correlation  $\dot{x}_2$  for different number data points with poly order 4

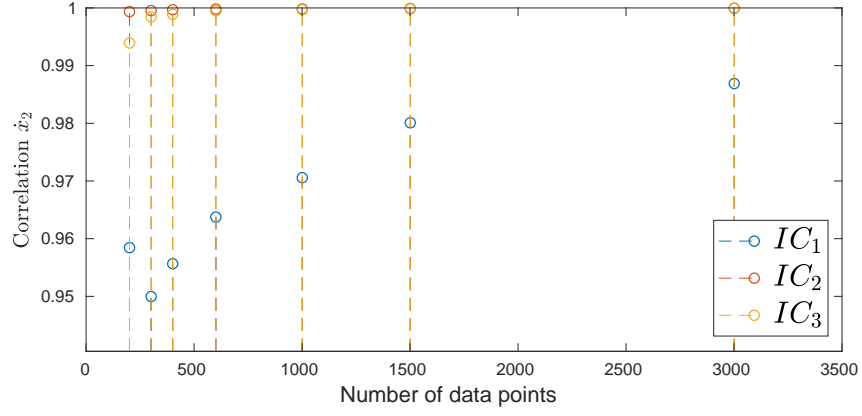
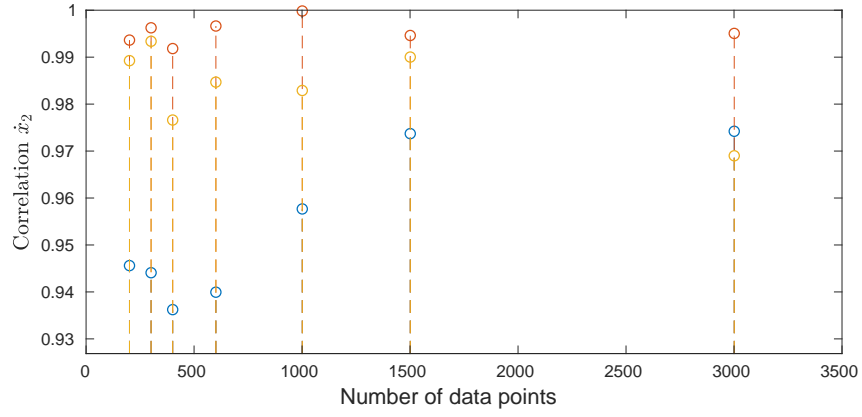
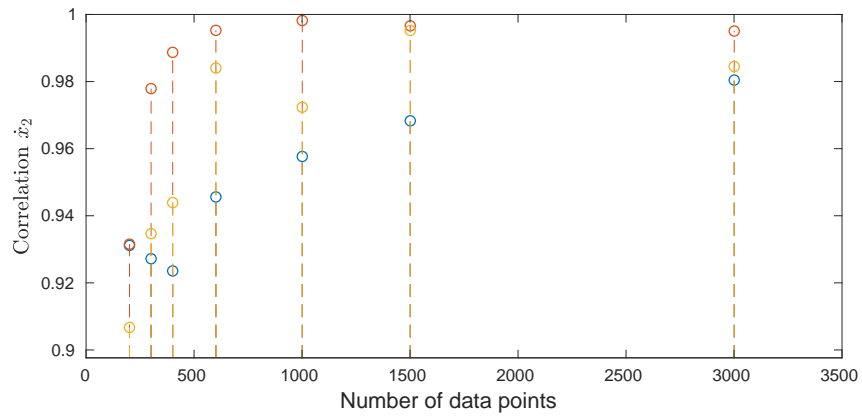
Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.94561	0.94409	0.93623	0.93994	0.95767	0.97371	0.97421
$IC_2$	0.99363	0.99625	0.99183	0.99663	0.99983	0.99462	0.99506
$IC_3$	0.98926	0.99338	0.97661	0.98468	0.98290	0.99002	0.96900

Table 9 - Correlation  $\dot{x}_2$  for different number data points with poly order 5

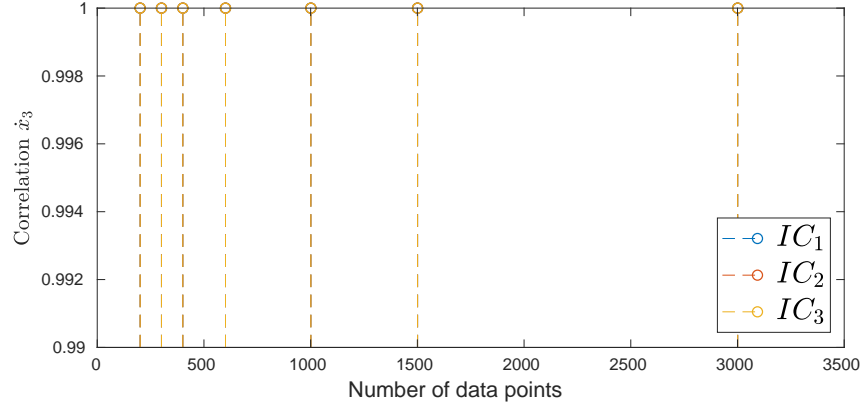
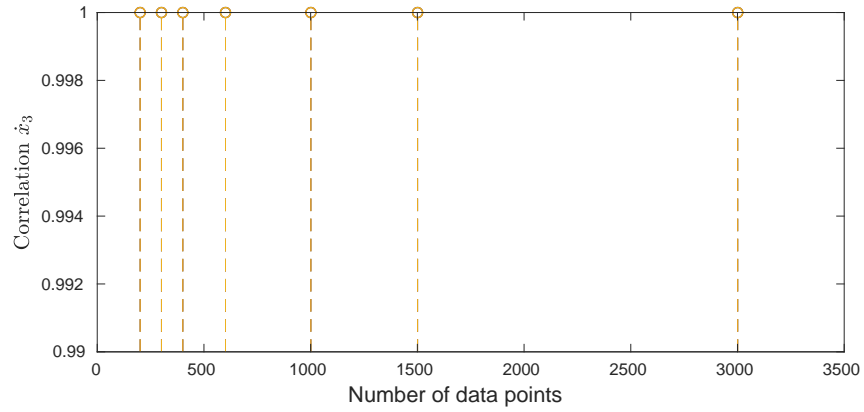
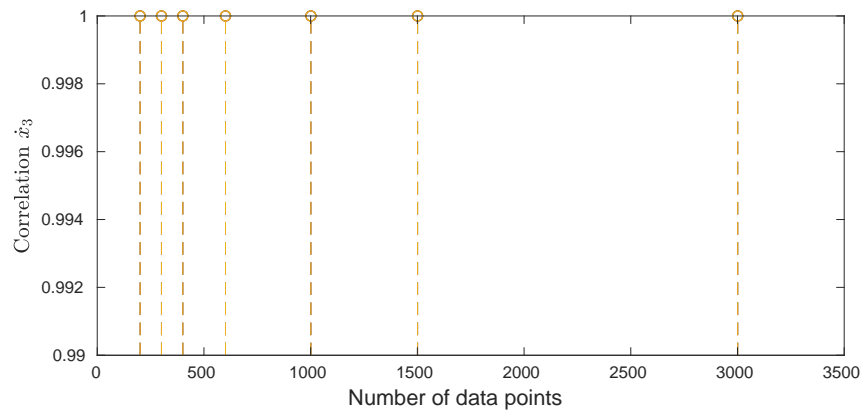
Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.93116	0.92719	0.92355	0.94562	0.95767	0.96830	0.98043
$IC_2$	0.93153	0.97789	0.9887	0.99527	0.99819	0.99661	0.99506
$IC_3$	0.90671	0.93467	0.94391	0.98403	0.97235	0.99524	0.98449

Table 10 - Correlation  $\dot{x}_3$  for different number data points with poly order 3

Data points	200	300	400	600	1000	1500	3000
$IC_1$	1	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1	1

Figure 17 - Correlation of  $\dot{x}_2$  with different number of data points(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: Correlation of the  $\dot{x}_2$  variable between the identified and original Duffing oscillator with different number of data points varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Figure 18 - Correlation of  $\dot{x}_3$  with different number of data points(a) Until 3<sup>rd</sup> polynomial order(b) Until 4<sup>th</sup> polynomial order(c) Until 5<sup>th</sup> polynomial order

Caption: Correlation of the  $\dot{x}_3$  variable between the identified and original Duffing oscillator with different number of data points varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 11 - Correlation  $\dot{x}_3$  for different number data points with poly order 4

Data points	200	300	400	600	1000	1500	3000
$IC_1$	1	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1	1

Table 12 - Correlation  $\dot{x}_3$  for different number data points with poly order 5

Data points	200	300	400	600	1000	1500	3000
$IC_1$	1	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1	1

Table 13 - Total correlation for different number of data points with poly order 3

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.99993	0.99992	0.99993	0.99994	0.99995	0.99997	0.99998
$IC_2$	1	1	1	1	1	1	1
$IC_3$	0.99999	1	1	1	1	1	1

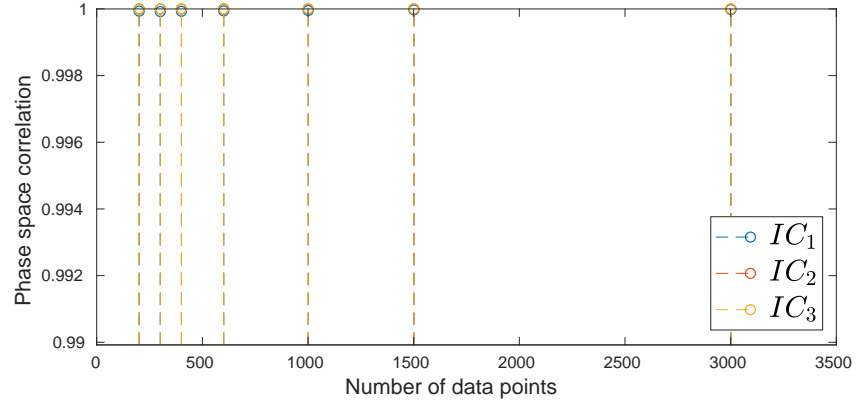
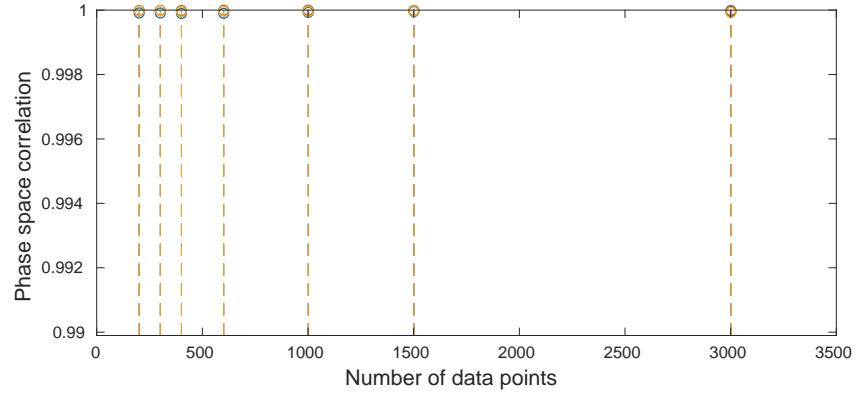
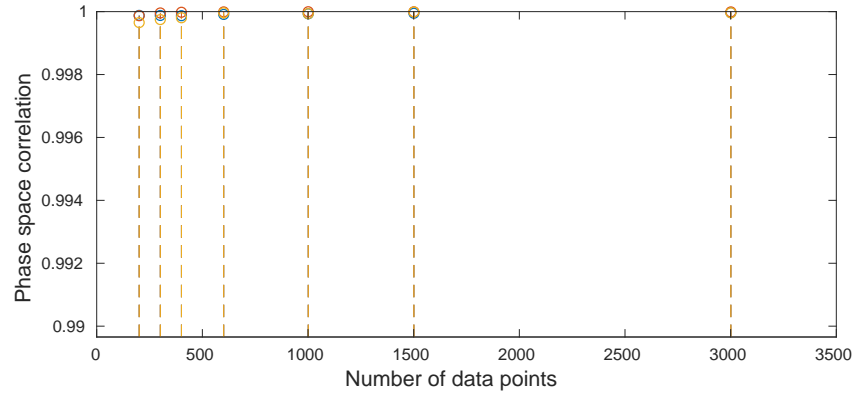
Table 14 - Total correlation for different number of data points with poly order 4

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.99991	0.99991	0.99990	0.99990	0.99993	0.99996	0.99996
$IC_2$	0.99999	0.99999	0.99999	0.99999	1	0.99999	0.99999
$IC_3$	0.99997	0.99999	0.99995	0.99996	0.99996	0.99998	0.99993

Table 15 - Total correlation for different number of data points with poly order 5

Data points	200	300	400	600	1000	1500	3000
$IC_1$	0.99988	0.99988	0.99987	0.99991	0.99993	0.99995	0.99997
$IC_2$	0.99987	0.99996	0.99998	0.99999	1	0.99999	0.99999
$IC_3$	0.99965	0.99975	0.99981	0.99996	0.99994	0.99999	0.99996

Figure 19 - Correlation of phase space with different number of data points

(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: Correlation of the phase space variable between the identified and original Duffing oscillator with different number of data points varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

were presented by separately examining each of the three dimensions of the Duffing system modeling, and the final analysis encompassed the phase space composed of these three dimensions. Overall, the  $\dot{x}_3$  dimension did not yield particularly interesting results, as observed in the graphs in Figure 18 and Tables 10, 11, and 12. This is because it is composed of only a constant term, mostly demonstrating a high correlation between the dynamics.

Analyzing the correlation results between the dynamics, as illustrated in Figures 16 to 19 and Tables 4 to 15, in most cases where the RMSE increased, the correlation decreased. For instance, between 200 and 300 data points for  $IC_1$  with third-order polynomial functions, there was a 12.71% increase in RMSE and a decrease in the correlation of  $\dot{x}_1$  by  $-0.00314$  or  $-0.32\%$  concerning the correlation with 200 data points. Similarly, for  $\dot{x}_2$ , there was a decrease of  $-0.00848$  or  $-0.88\%$ . There was only one comparison where the correlation value increased as the RMSE value rose; the correlation increased by  $0.0029$  or  $0.30\%$  when the RMSE increased by  $6.66\%$ . Although there were also cases where the correlation decreased as the RMSE decreased, comparing simulations with 3000 data points to those with 200 data points, the correlation increased, except for cases with polynomials up to the fourth order for  $IC_3$  due to the presence of an outlier. However, like the RMSE, when compared to 1500 data points, the dynamics exhibited higher correlation.

The increase in correlation as the number of data points rises is further supported by the phase space correlation, depicted in Tables 13, 14, and 15, as well as Figure 19. It is also quite evident how simulations with only polynomial functions up to the third order in the library exhibit higher correlation with the numerical data. Although the correlation values are higher for simulations with 3000 data points when there are polynomial functions up to the fifth order compared to those up to the fourth order, for the first three quantities 200, 300, and 400 data points those with functions up to the fourth order have higher correlation values. Similar to RMSE, it is not possible to determine how adding more functions will affect the result. However, when comparing all correlation and RMSE results with polynomial functions up to the third degree, it is noted that the compared dynamics have better values than in the other two cases. It is also safe to state that increasing the number of data points can indeed improve results, even though an exact relationship between the number of points and the result improvement is not observed.

### 3.7.2 Noise intensity

In this case, we test how the quality of the available data will affect the identified dynamics. For this purpose, seven different intensities of Gaussian noise were selected,

Table 16 - RMSE for different noise intensity with poly order 3

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	$2.6532e^{-14}$	0.02817	0.049328	0.08883	0.14142	0.21762	0.24869
$IC_2$	$3.062e^{-12}$	0.33483	0.57198	0.8484	1.1206	1.249	1.4121
$IC_3$	$3.3498e^{-14}$	0.026759	0.048535	0.08527	0.14134	0.1791	0.26094

Table 17 - RMSE for different noise intensity with poly order 4

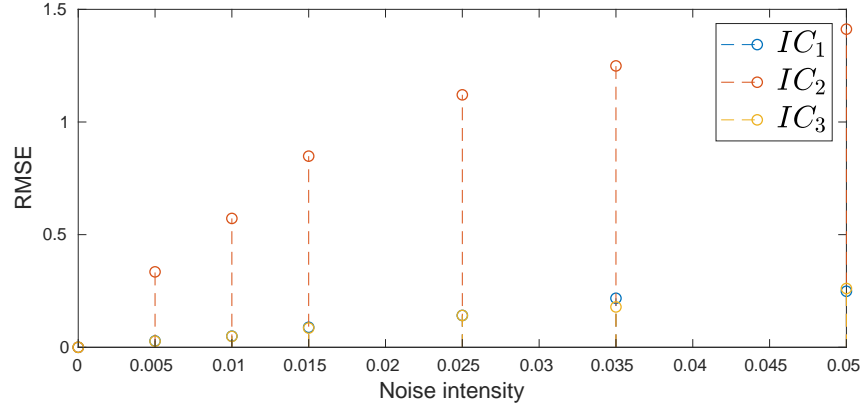
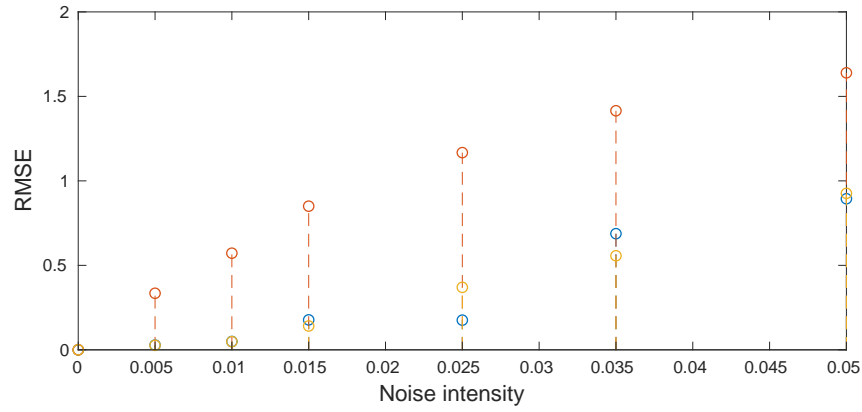
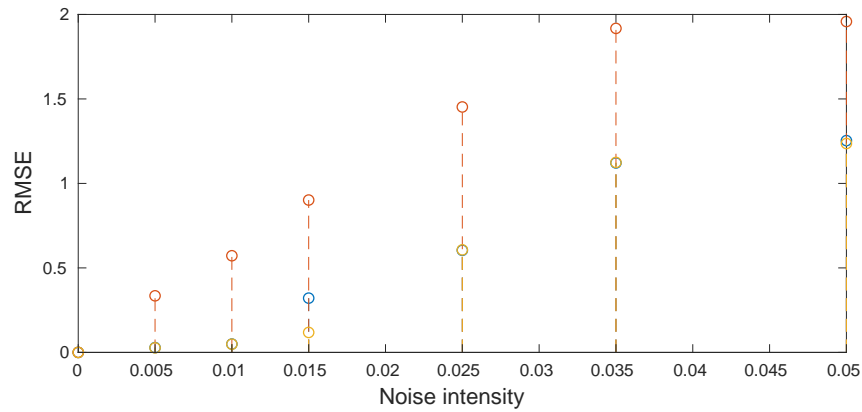
Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	$3.0979e^{-14}$	0.02817	0.049328	0.17735	0.17572	0.68755	0.89416
$IC_2$	$3.0367e^{-12}$	0.33483	0.57198	0.85026	1.1672	1.4149	1.6394
$IC_3$	$2.9903e^{-14}$	0.026759	0.048535	0.14118	0.36987	0.55666	0.92621

namely 0, 0.005, 0.010, 0.015, 0.025, 0.035, and 0.050. The time interval for obtaining the samples was from 0 to 30 with 601 evenly distributed samples. All other parameters remained the same as in the simulations, with the only change being the number of data points, except for  $IC_3$ , which was set as  $[2, -1, 0]$ . This adjustment was made because, in some of the 100 simulations with a noise intensity of 0.050, the method was unable to identify the dynamics, resulting in the identification of many polynomial terms. The MATLAB code was unable to calculate the dynamics until the time of 100 to measure the RMSE and correlation.

Examining the results in Tables 16, 17, and 18, as well as Figure 20, it was observed that as the noise intensity increased, the RMSE also increased for all simulations, except for  $IC_1$  simulations with polynomials up to the fourth order and a noise intensity of 0.025, where the decrease was only  $-0.00163$  or  $-0.92\%$ . Also noteworthy were the results for a noise intensity of 0, where the RMSE value could be considered computationally zero. This was expected based on the results of Lopes and Cunha Jr (2022), as at a noise intensity of 0, the differences in momentum and conservation of energy between the dynamics were also numerically zero. Another interesting pattern to observe was that for noise intensities of 0.005 and 0.010, regardless of the number of polynomial functions used, the RMSE values were the same. Only from 0.015 onwards did the RMSE increase as the number of polynomial functions increased. The only exception was at a noise intensity of 0.015 between fourth and fifth-order polynomial functions, showing a decrease of  $-0.02313$  or  $-16.38\%$ .

Analyzing the results of correlation, we observe Tables 19 to 30 and Figures 21 to 24. It becomes evident that noise intensity has a negative impact on the inferred dynamics' quality. Starting with an analysis of the correlation in dimension  $x_1$  with polynomials up to the third order, it's clear that increasing noise intensity reduces the correlation between dynamics. The most significant difference is observed for  $IC_2$ , with

Figure 20 - RMSE with different intensity noise

(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: RMSE between the identified and original Duffing oscillator with different intensity noise varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuations on the noise.



Table 18 - RMSE for different noise intensity with poly order 5

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	$2.6532e^{-14}$	0.02817	0.049328	0.32106	0.60406	1.121	1.2533
$IC_2$	$3.062e^{-12}$	0.33483	0.57198	0.90164	1.4525	1.9175	1.9581
$IC_3$	$3.3498e^{-14}$	0.026759	0.048535	0.11805	0.60669	1.1241	1.2361

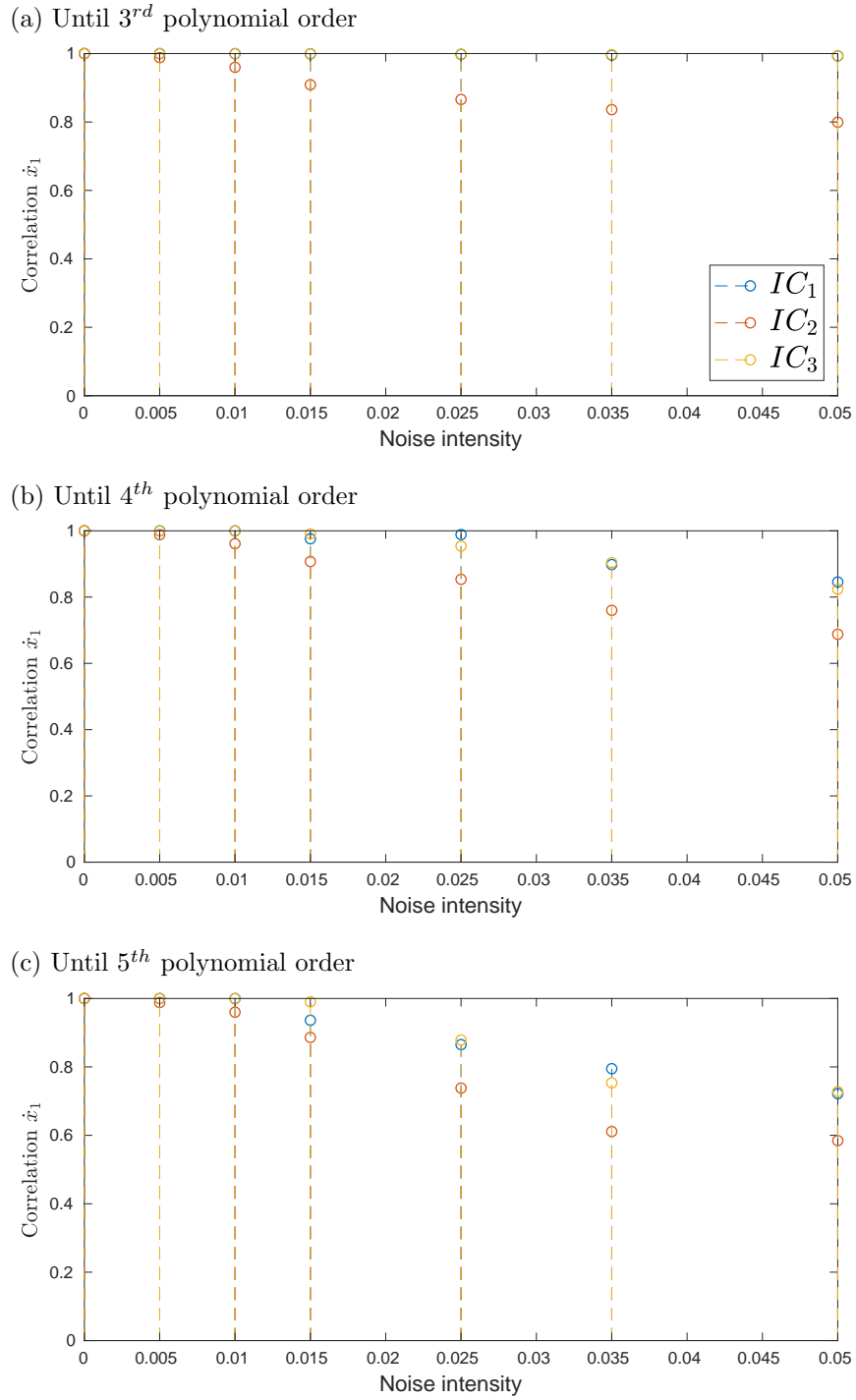
Table 19 - Correlation of  $\dot{x}_1$  for different noise intensity with poly order 3

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99991	0.99975	0.99917	0.99785	0.99539	0.9931
$IC_2$	1	0.9882	0.95993	0.90895	0.86586	0.83622	0.79942
$IC_3$	1	0.99993	0.99975	0.99932	0.99806	0.99684	0.99303

a drop of 0.20058 or  $-19.10\%$  between the lowest and highest noise levels. This pattern repeats for simulations with more functions in the library, showing a decreasing correlation as noise levels increase. For function libraries with polynomials up to the fourth and fifth orders, there are correlation drops of 0.31226 and 0.4156, respectively, for  $IC_2$ .

Now, comparing the same noise levels but with different numbers of functions, we observe a more distinct percentage difference compared to simulations varying the quantity of data. For instance, the correlation values show drops of  $-14.86\%$ ,  $-13.97\%$ , and  $-17.02\%$  when comparing the correlation of noise intensity 0.05 with polynomials up to the fourth order compared to up to the third order. The same comparison, but now between fifth-order polynomials compared to fourth-order ones, shows drops of  $-14.63\%$ ,  $-15.03\%$ , and  $-11.70\%$ . Comparing the tests with more functions to those with fewer functions, the drops are  $-37.60\%$ ,  $-36.79\%$ , and  $-36.48\%$ . Achieving an exact value would require a very large number of simulations due to the use of different random generations of Gaussian numbers, but the close values indicate a stronger relationship than in simulations with varying data points.

As for the correlation results for dimensions  $\dot{x}_2$  and  $\dot{x}_3$ , they do not present any new findings that have not been highlighted before. Dimension  $\dot{x}_2$  shows a behavior similar to  $\dot{x}_1$ , as discussed earlier, while  $\dot{x}_3$  mostly has values very close to 1, similar to simulations with varying data points. The same can be said for the total correlation. In this set of simulations, it emphasizes again how lower noise intensity and fewer polynomial functions yield better results. These results highlight the importance of capturing high-quality sample results to feed into the SINDy method. The lower the quality, the results also demonstrate that greater knowledge of possible functions and avoiding excessive quantities of functions can contribute to a better outcome.

Figure 21 - Correlation of  $\dot{x}_1$  with different intensity noise

Caption: Correlation of the  $\dot{x}_1$  variable between the identified and original Duffing oscillator with different intensity noise varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 20 - Correlation of  $x_1$  for different noise intensity with poly order 4

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99991	0.99975	0.97598	0.98863	0.89817	0.84548
$IC_2$	1	0.9882	0.96118	0.90727	0.85358	0.76006	0.68774
$IC_3$	1	0.99993	0.99975	0.99044	0.95379	0.90387	0.82401

Table 21 - Correlation of  $x_1$  for different noise intensity with poly order 5

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99991	0.99975	0.93619	0.86521	0.7948	0.72175
$IC_2$	1	0.9882	0.95993	0.88631	0.73815	0.61111	0.5844
$IC_3$	1	0.99993	0.99975	0.99024	0.87852	0.7534	0.7276

Table 22 - Correlation of  $x_2$  for different noise intensity with poly order 3

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99989	0.99969	0.99897	0.99734	0.99431	0.99151
$IC_2$	1	0.98151	0.94512	0.8858	0.83061	0.79754	0.7566
$IC_3$	1	0.99992	0.99969	0.99916	0.99761	0.99611	0.99147

Table 23 - Correlation of  $x_2$  for different noise intensity with poly order 4

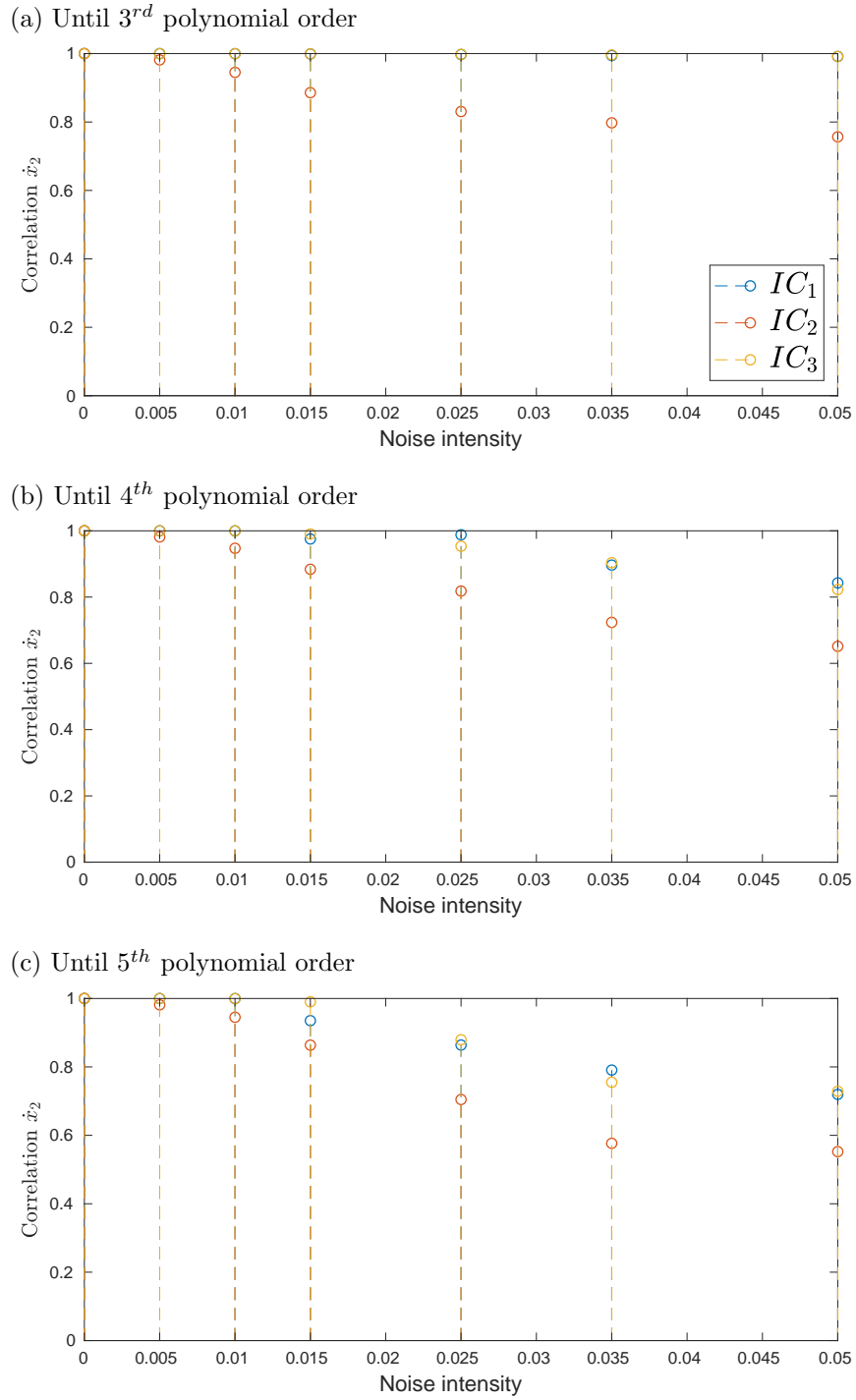
Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99989	0.99969	0.97538	0.98803	0.8965	0.84284
$IC_2$	1	0.98151	0.94728	0.88387	0.81821	0.72374	0.6514
$IC_3$	1	0.99992	0.99969	0.99036	0.95368	0.9037	0.82282

Table 24 - Correlation of  $x_2$  for different noise intensity with poly order 5

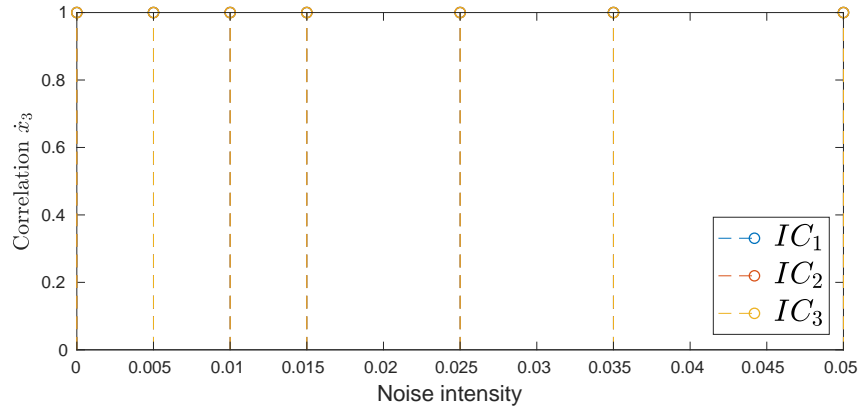
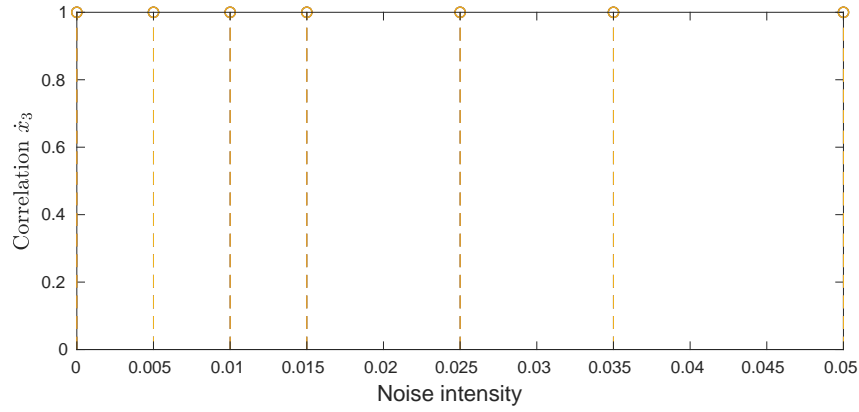
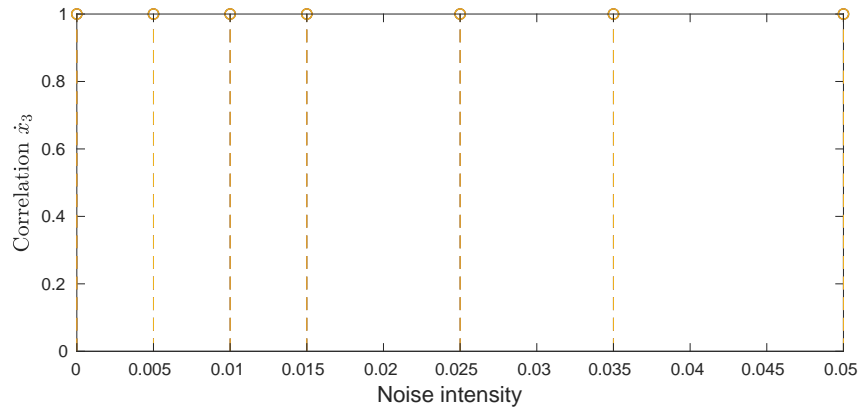
Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	0.99989	0.99969	0.93521	0.86393	0.79073	0.71966
$IC_2$	1	0.98151	0.94512	0.86362	0.70472	0.57664	0.55231
$IC_3$	1	0.99992	0.99969	0.99019	0.87955	0.75517	0.72871

Table 25 - Correlation of  $x_3$  for different noise intensity with poly order 3

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1	1

Figure 22 - Correlation of  $\dot{x}_2$  with different intensity noise

Caption: Correlation of the  $\dot{x}_2$  variable between the identified and original Duffing oscillator with different intensity noise varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Figure 23 - Correlation of  $\dot{x}_3$  with different intensity noise(a) Until 3<sup>rd</sup> polynomial order(b) Until 4<sup>th</sup> polynomial order(c) Until 5<sup>th</sup> polynomial order

Caption: Correlation of the  $\dot{x}_3$  variable between the identified and original Duffing oscillator with different intensity noise varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 26 - Correlation of  $x_3$  for different noise intensity with poly order 4

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	1	1	0.99998	0.99998
$IC_2$	1	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1	1

Table 27 - Correlation of  $x_3$  for different noise intensity with poly order 5

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	1	1	0.99995	0.99998
$IC_2$	1	1	1	1	1	0.99996	0.99997
$IC_3$	1	1	1	1	1	0.99996	1

Table 28 - Total correlation for different noise intensity with poly order 3

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	1	1	0.99999	0.99999
$IC_2$	1	0.99997	0.99992	0.99983	0.99975	0.9997	0.99964
$IC_3$	1	1	1	1	1	0.99999	0.99998

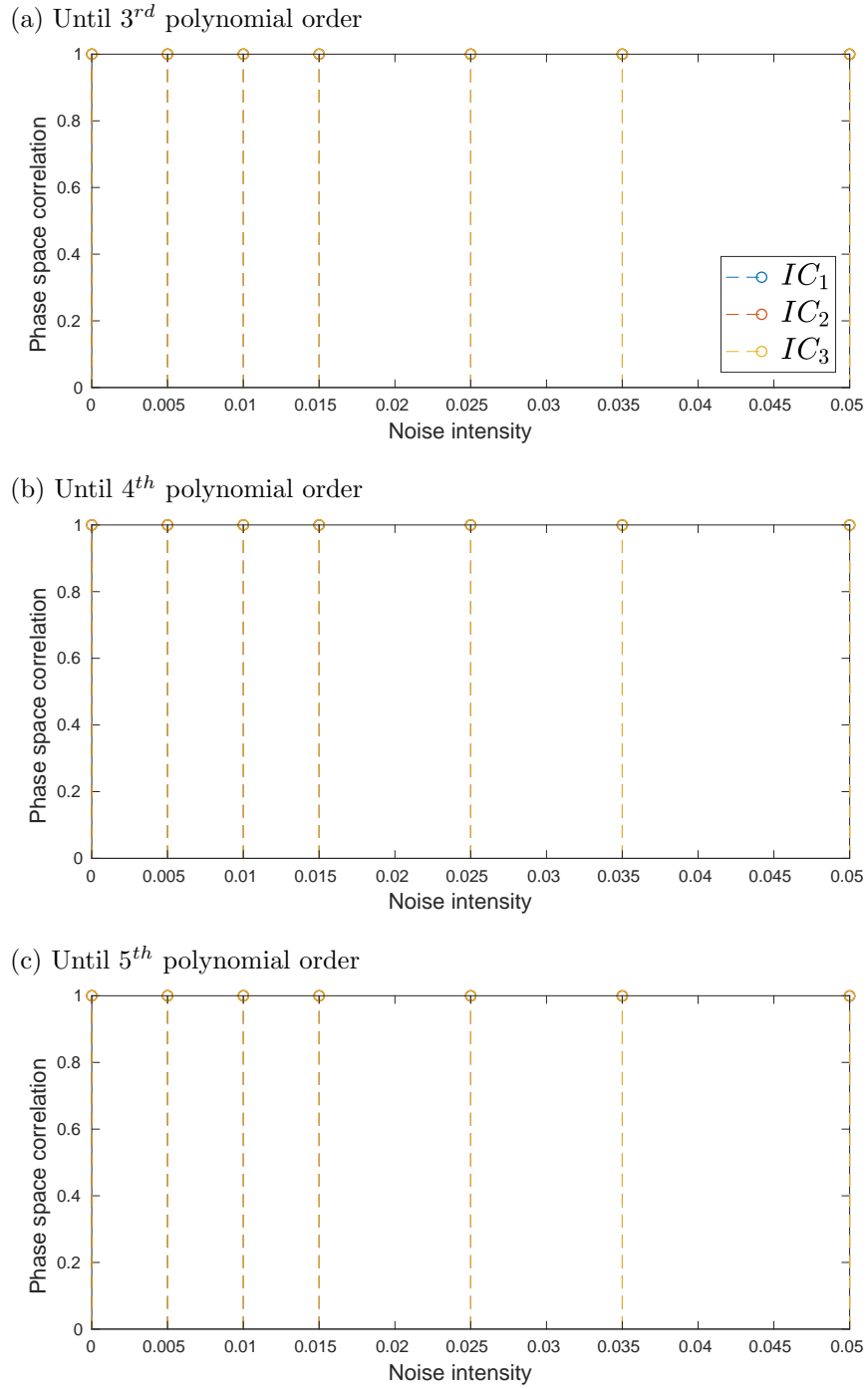
Table 29 - Total correlation for different noise intensity with poly order 4

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	0.99996	0.99998	0.9998	0.99971
$IC_2$	1	0.99997	0.99992	0.99983	0.99973	0.9996	0.99949
$IC_3$	1	1	1	0.99998	0.99991	0.99982	0.99967

Table 30 - Total correlation for different noise intensity with poly order 5

Noise int.	0	0.005	0.010	0.015	0.025	0.035	0.05
$IC_1$	1	1	1	0.99989	0.99976	0.99961	0.99951
$IC_2$	1	0.99997	0.99992	0.9998	0.99956	0.99935	0.9993
$IC_3$	1	1	1	0.99998	0.99978	0.99953	0.99949

Figure 24 - Correlation of phase space with different intensity noise



Caption: Correlation of the phase space variable between the identified and original Duffing oscillator with different intensity noise varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 31 - RMSE for different time interval with poly order 3

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.19712	0.20817	0.23268	0.21787	0.20472	0.20858
$IC_2$	0.015755	0.01628	0.017698	0.014526	0.014124	0.01586
$IC_3$	0.014797	0.019587	0.018819	0.01819	0.021511	0.018593

### 3.7.3 Time interval

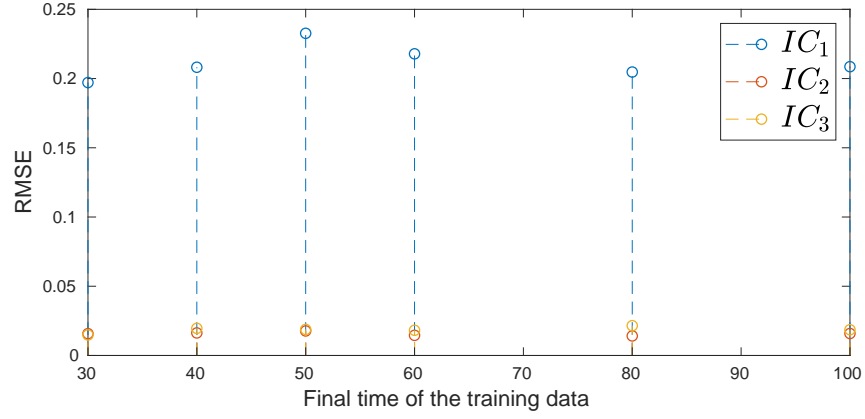
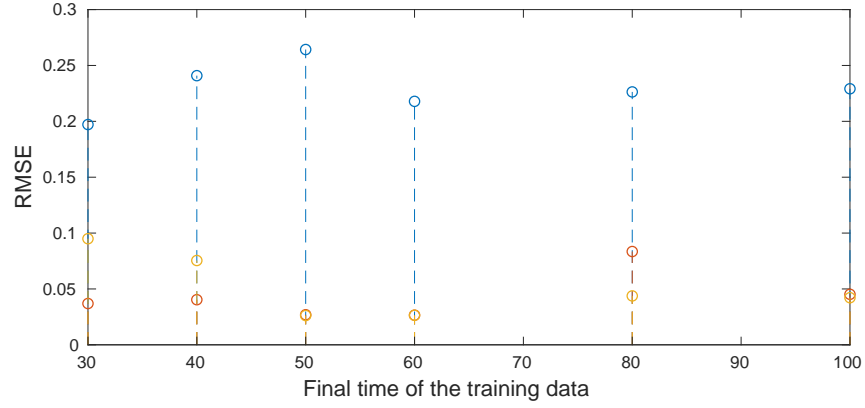
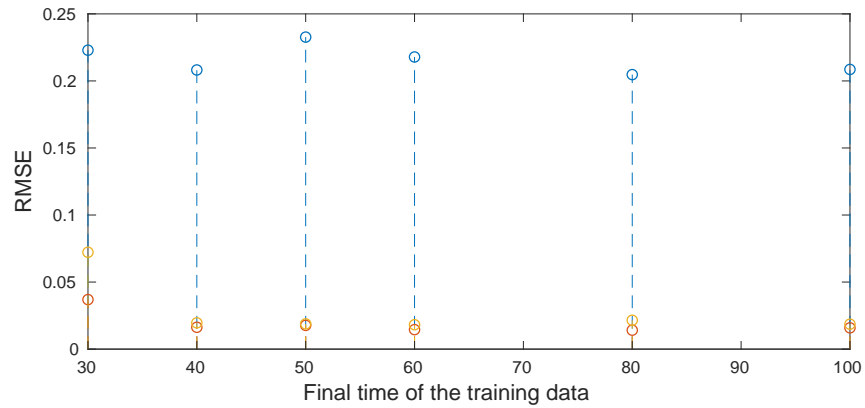
Finally, we conduct a series of simulations where we determine various time intervals with a fixed number of 1000 data points, specifically, intervals of 30, 40, 50, 60, 80, and 100. All other parameters remain consistent with the test set that varies noise intensity, except for the fixed noise intensity at 0.005 in this simulation.

Upon examining Figure 25 and Tables 31,32, and 33, which present the results of the RMSE calculation, it is evident that, unlike the first two sets of simulations, there is no clear relationship between the increase in data capture time and the quality of the inferred dynamics. This, in turn, affects the RMSE value. For example, when looking at the results for  $IC_1$  with third-order polynomial functions in the first row of Table 31, there is an increase in RMSE of 0.01105 or 5.61% between 30 and 40 time intervals. From 40 to 50, there is an increase of 0.02451 or 11.77%. Between 50 and 60, there is a decrease of  $-0.01481$  or  $-6.36\%$ , followed by another decrease of  $-0.01315$  or  $-6.04\%$  from 60 to 80. Finally, there is an increase of 0.00386 or 1.89%. Comparing the largest and smallest data collection time intervals, we observe an increase of only 0.01146 or 5.81%. This lack of a clear tendency is present for all initial conditions across the three different numbers of polynomial functions in the library, demonstrating no direct relationship between the improvement of RMSE in the inferred dynamics and an increase in the time interval.

However, when comparing values between different simulations with various function libraries, we find that all RMSE values with polynomials up to the fourth order are higher than those up to the third. On the other hand, those with fifth-order polynomials mostly decrease when compared to simulations up to the fourth order, similar to the results observed in simulations with different numbers of data points. The most interesting result emerges when comparing values between simulations with polynomial libraries up to the third and fifth orders. We observe that, for the data collection time interval up to 30, the RMSE values for simulations with more functions are higher. However, after this point, all values remain consistent. As the random number generation is fixed with the seed, and when changing the number of polynomial functions in the library, the random number generation is restarted. This may indicate that for these specific simulations, the time interval of 40 is the ideal value, and increasing the interval further will not affect the result's quality.



Figure 25 - RMSE with different time interval

(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: RMSE between the identified and original Duffing oscillator with different time interval varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuations on the noise.

Table 32 - RMSE for different time interval with poly order 4

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.19712	0.24087	0.26427	0.21787	0.22632	0.22914
$IC_2$	0.03695	0.040366	0.026913	0.026515	0.083481	0.045199
$IC_3$	0.094949	0.075452	0.026174	0.02644	0.043767	0.04218

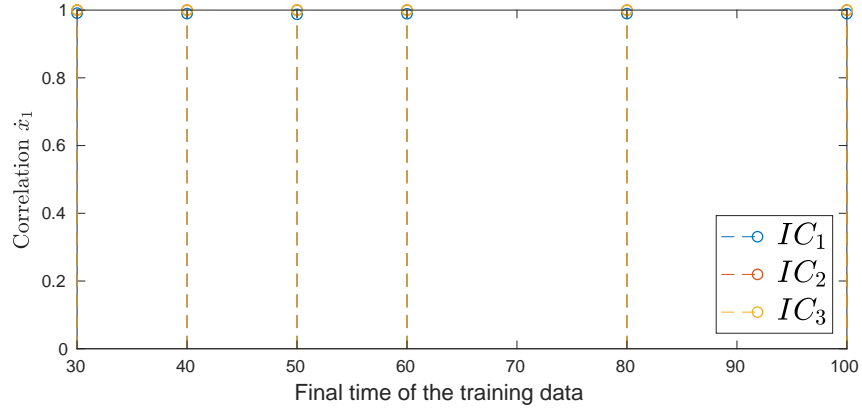
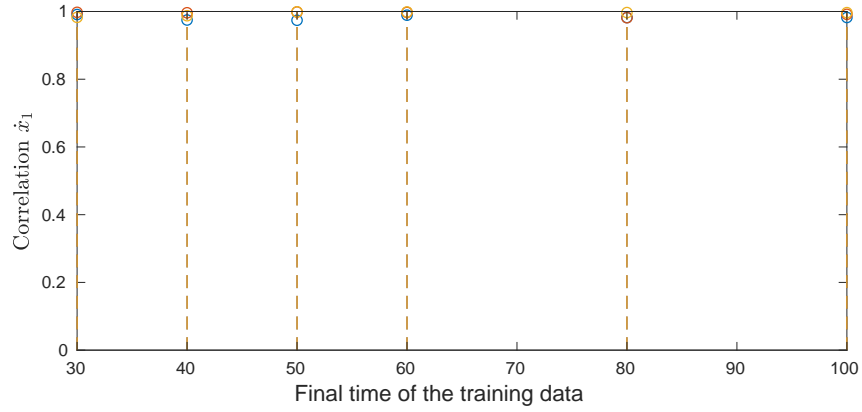
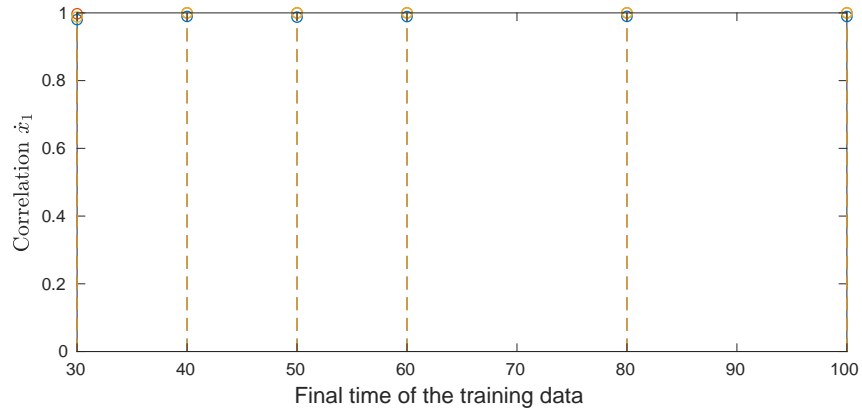
Table 33 - RMSE for different time interval with poly order 5

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.22293	0.20817	0.23268	0.21787	0.20472	0.20858
$IC_2$	0.036932	0.01628	0.017698	0.014526	0.014124	0.01586
$IC_3$	0.072229	0.019587	0.018819	0.01819	0.021511	0.018593

Upon examining the correlation results across different dimensions, illustrated in Figures 26, 27, 28, and 29, as well as detailed in Tables 34 to 45, it becomes apparent that the extension of the data capture time interval has no discernible impact on the quality of the results. The most significant relative difference between the longest and shortest intervals led to a mere 2.19% correlation increase, observed in  $IC_3$  for the correlation of the  $\dot{x}_2$  dimension with polynomials up to the fourth order. Comparing identical simulations but augmenting the number of functions similarly revealed no disparity in the RMSE results. A marginal decrease in value is noted when contrasting correlations with polynomials up to the fourth order against those up to the third order. Likewise, in the comparison of correlations with polynomials up to the fourth order versus those with fifth-order polynomials, most exhibit a correlation closer to 1 in the library composed of polynomials up to the fifth order. Hence, the results suggest that augmenting the data capture time interval does not equate to an enhancement, as the smallest interval already furnishes adequate data for inferring the dynamics.

Table 34 - Correlation of  $\dot{x}_1$  for different time interval with poly order 3

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.99103	0.98957	0.98751	0.9891	0.98965	0.9894
$IC_2$	0.99993	0.99993	0.99993	0.99995	0.99995	0.99994
$IC_3$	0.99994	0.99991	0.99992	0.99992	0.9999	0.99991

Figure 26 - Correlation of  $\dot{x}_1$  with different time interval(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: Correlation of the  $\dot{x}_1$  variable between the identified and original Duffing oscillator with different time interval varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 35 - Correlation of  $\dot{x}_1$  for different time interval with poly order 4

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.99103	0.97487	0.97413	0.9891	0.98218	0.9823
$IC_2$	0.99748	0.99619	0.99838	0.99728	0.98167	0.99156
$IC_3$	0.98272	0.98752	0.99901	0.9986	0.99681	0.99657

Table 36 - Correlation of  $\dot{x}_1$  for different time interval with poly order 5

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.98009	0.98957	0.98751	0.9891	0.98965	0.9894
$IC_2$	0.99747	0.99993	0.99993	0.99995	0.99995	0.99994
$IC_3$	0.98795	0.99991	0.99992	0.99992	0.9999	0.99991

Table 37 - Correlation of  $\dot{x}_2$  for different time interval with poly order 3

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.97638	0.97246	0.96712	0.97124	0.97273	0.97209
$IC_2$	0.99985	0.99986	0.99985	0.99989	0.9999	0.99987
$IC_3$	0.99988	0.99981	0.99982	0.99983	0.99979	0.99981

Table 38 - Correlation of  $\dot{x}_2$  for different time interval with poly order 4

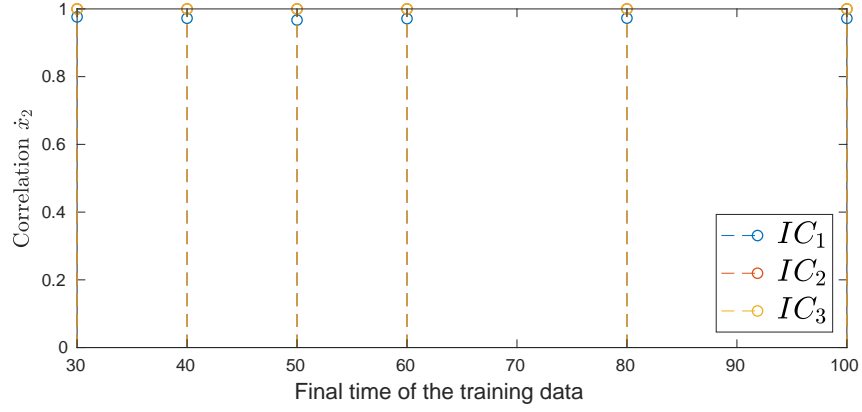
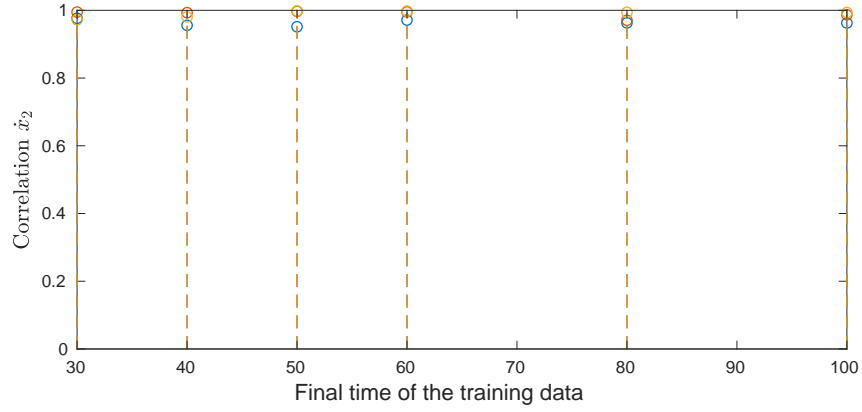
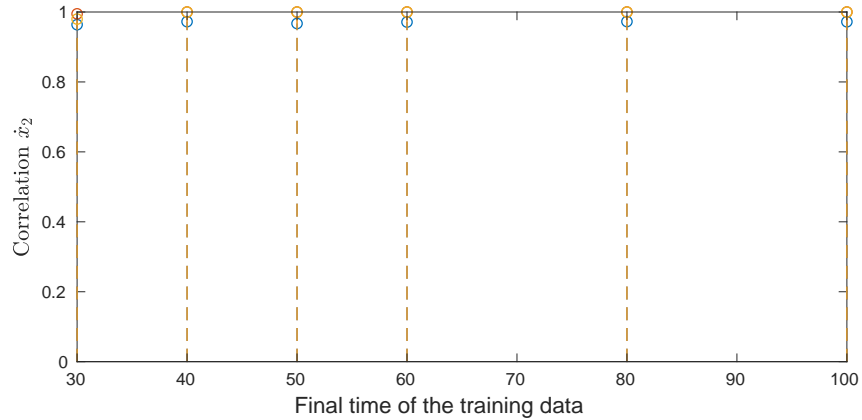
$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.97638	0.95568	0.95149	0.97124	0.96298	0.96267
$IC_2$	0.9947	0.99305	0.99696	0.99529	0.97065	0.98675
$IC_3$	0.97219	0.98053	0.99809	0.99754	0.994	0.99351

Table 39 - Correlation of  $\dot{x}_2$  for different time interval with poly order 5

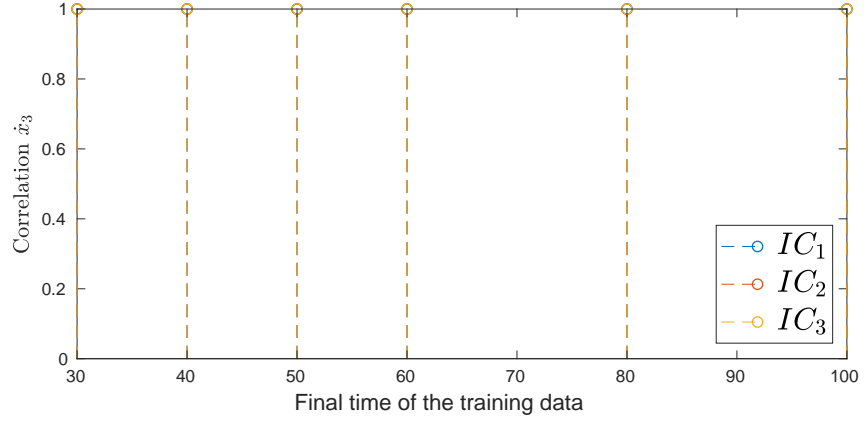
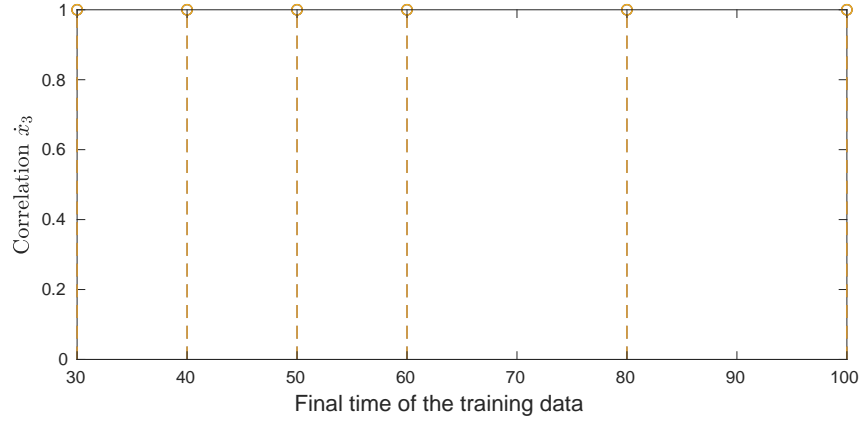
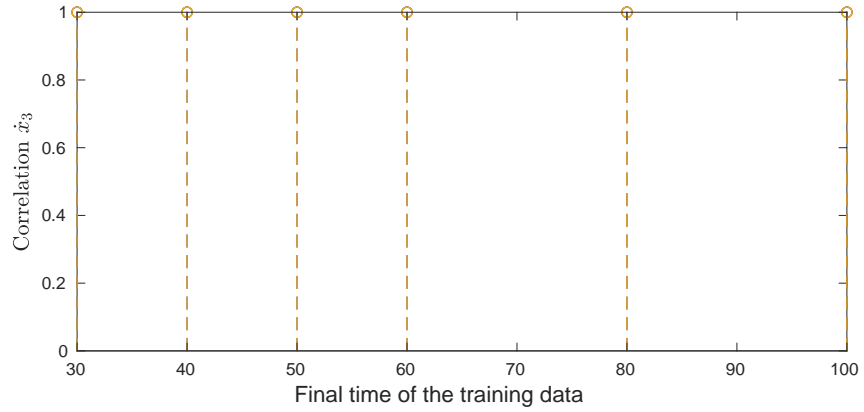
$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.96375	0.97246	0.96712	0.97124	0.97273	0.97209
$IC_2$	0.99474	0.99986	0.99985	0.99989	0.9999	0.99987
$IC_3$	0.97976	0.99981	0.99982	0.99983	0.99979	0.99981

Table 40 - Correlation of  $\dot{x}_3$  for different time interval with poly order 3

$\Delta$ time	30	40	50	60	80	100
$IC_1$	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1

Figure 27 - Correlation of  $\dot{x}_2$  with different time interval(a) Until  $3^{rd}$  polynomial order(b) Until  $4^{th}$  polynomial order(c) Until  $5^{th}$  polynomial order

Caption: Correlation of the  $\dot{x}_2$  variable between the identified and original Duffing oscillator with different time interval varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Figure 28 - Correlation of  $\dot{x}_3$  with different time interval(a) Until 3<sup>rd</sup> polynomial order(b) Until 4<sup>th</sup> polynomial order(c) Until 5<sup>th</sup> polynomial order

Caption: Correlation of the  $\dot{x}_3$  variable between the identified and original Duffing oscillator with different time intensity varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.

Table 41 - Correlation of  $\dot{x}_3$  for different time interval with poly order 4

$\Delta$ time	30	40	50	60	80	100
$IC_1$	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1

Table 42 - Correlation of  $\dot{x}_3$  for different time interval with poly order 5

$\Delta$ time	30	40	50	60	80	100
$IC_1$	1	1	1	1	1	1
$IC_2$	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1

Table 43 - Total correlation for different time interval with poly order 3

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.99996	0.99996	0.99995	0.99995	0.99996	0.99996
$IC_2$	1	1	1	1	1	1
$IC_3$	1	1	1	1	1	1

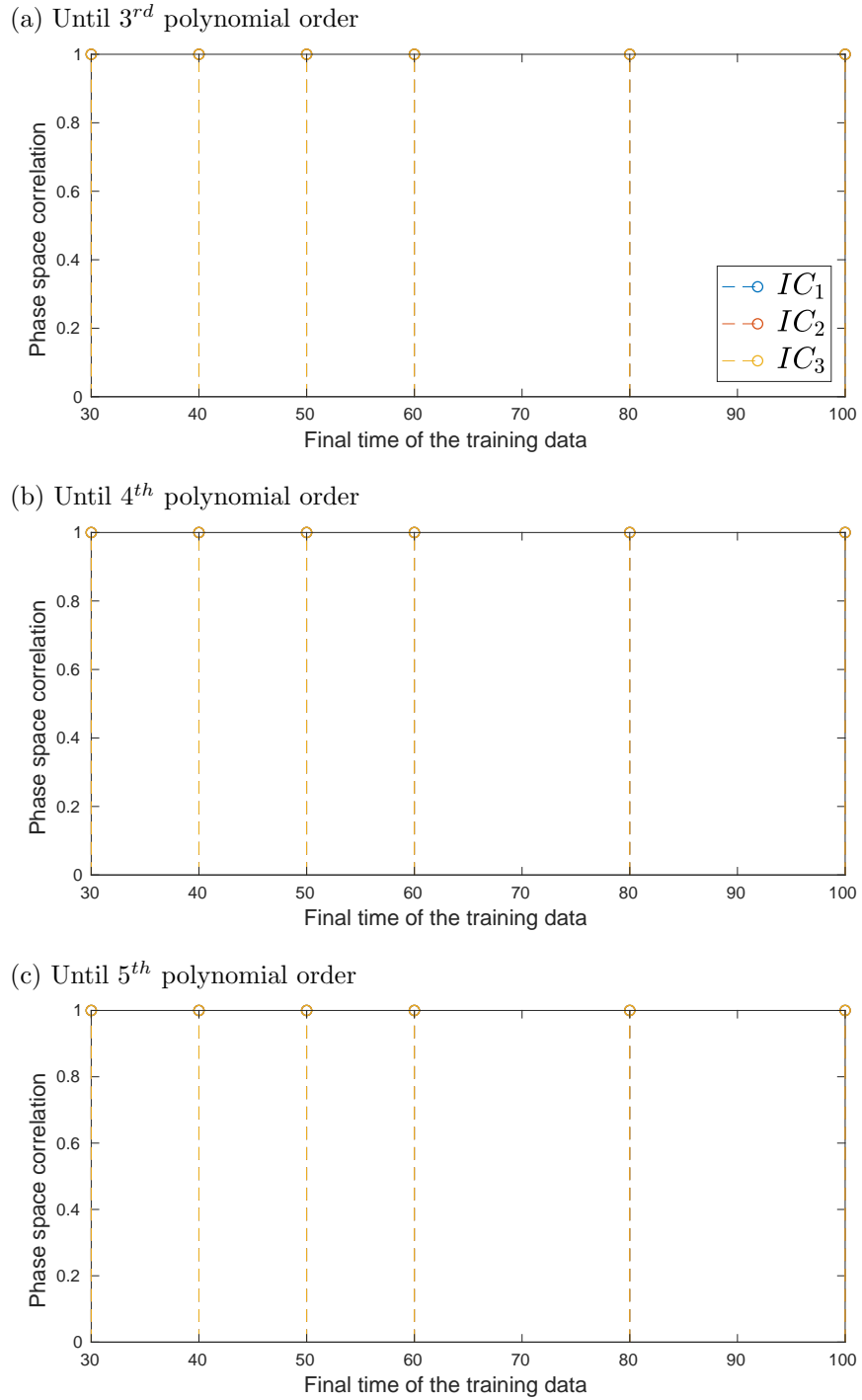
Table 44 - Total correlation for different time interval with poly order 4

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.99996	0.99993	0.99992	0.99995	0.99994	0.99994
$IC_2$	0.99999	0.99999	0.99999	0.99999	0.99995	0.99998
$IC_3$	0.99995	0.99996	1	1	0.99999	0.99999

Table 45 - Total correlation for different time interval with poly order 5

$\Delta$ time	30	40	50	60	80	100
$IC_1$	0.99994	0.99996	0.99995	0.99995	0.99996	0.99996
$IC_2$	0.99999	1	1	1	1	1
$IC_3$	0.99996	1	1	1	1	1

Figure 29 - Correlation of phase space with different time interval



Caption: Correlation of the phase space variable between the identified and original Duffing oscillator with different time intensity varying the number of candidate functions, each result is the mean of one hundred tests with changes on the fluctuation on the noise.



## CONCLUSIONS

In summary, this study comprehensively presented the Sparse Identification of Nonlinear Dynamics (SINDy) method, showcasing its application across diverse dynamic systems to rigorously test its robustness and efficacy in inferring chaotic systems. The method demonstrated exceptional effectiveness in accurately capturing the underlying dynamics, even in chaotic scenarios. An examination of a trigonometric system, incorporating solely polynomial functions in the candidate function library, revealed intriguing results. While the Taylor series expansion of the trigonometric function was inferred, an additional dissipative polynomial term surfaced. Attempts to attribute this spurious term to the dissipation of the ODE45 MATLAB integrator were disproven through the utilization of an alternative integration method (ODE78), yielding identical results for both datasets.

To validate the method further, two distinct dynamic systems, the Van der Pol oscillator and the Rössler system, were employed. SINDy successfully reconstructed the original dynamic systems from the provided data, even accurately capturing the shape of attractors in simulations with chaotic dynamics. The investigation into the impact on dynamic proximity, by varying fundamental parameters in both the data and SINDy parameters, uncovered valuable insights. Three alterations directly affecting the captured data points from the Duffing oscillator’s numerical dynamic simulation were examined. Increasing the number of data points demonstrated a consistent trend of decreasing RMSE and increasing correlation, albeit with sporadic deviations attributed to Gaussian noise randomness. Notably, simulations with fewer unnecessary functions in the library outperformed those with more functions, emphasizing the importance of precision in the selection of candidate functions.

Similarly, varying the intensity of Gaussian noise underscored its predictable influence on inferred dynamics, reaffirming that lower data capture quality adversely impacts results. The analyses of datasets with the same number of data points but different time intervals revealed nuanced fluctuations in RMSE and correlation, indicating no significant improvement or deterioration with varying time intervals.

This extensive exploration positions the SINDy method as a compelling choice for applications across diverse scientific domains, particularly in mechanical systems such as the oscillatory dynamics of the Duffing oscillator. Furthermore, the findings emphasize the pivotal role of data precision, with an increased number of samples showcasing potential positive impacts on inferred dynamics. The study underscores how uncertainties in knowledge about fundamental laws negatively impact results, advocating for streamlined libraries with fewer unnecessary mathematical functions. Lastly, the demonstrated insensitivity of the method to prolonged data capture intervals suggests that the optimal

timeframe for data collection is crucial for accurate dynamics inference.

In summary, this study intricately elucidates the SINDy method in a didactic approach, employing diverse dynamical systems as benchmarks. The accompanying MATLAB codes, available on GitHub via the link ([https://github.com/DiegoMSL/dissertation\\_codes.git](https://github.com/DiegoMSL/dissertation_codes.git)), make it particularly accessible for beginners keen on grasping the intricacies of the method. Throughout the master's program, several materials on the method were disseminated, including presentations at prestigious national and international conferences such as VETOMAC XV, DINCON XIV, and the 14th WCCM-ECCOMAS (LOPES; Cunha Jr, 2019; LOPES; Cunha Jr, 2022; MATOS; JR, 2021). Additionally, a book chapter emerged from a presentation at the international conference NODYCON (LOPES; Cunha Jr, 2022). Looking ahead, future research endeavors could entail the application of the method to more intricate systems and real-world datasets, such as those derived from a Duffing oscillator or epidemiological phenomena like COVID-19.

## REFERENCES

- ALMEIDA, E. F. de; SILVA, S. da; JR, A. C. Physics-informed neural networks for solving elasticity problems. In: *27th International Congress on Mechanical Engineering (COBEM 2023)*. [S.l.: s.l.], 2023.
- APSEMIDIS, A.; PSARAKIS, S.; MOGUERZA, J. M. A review of machine learning kernel methods in statistical process monitoring. *Computers & Industrial Engineering*, vol. 142, p. 106376, 2020. ISSN 0360-8352.
- BADDOO, P. J. et al. Kernel learning for robust dynamic mode decomposition: linear and nonlinear disambiguation optimization. *Proceedings of the Royal Society A*, The Royal Society, vol. 478, no. 2260, p. 20210830, 2022.
- BETANCOURT-MAR, J. A.; ALARCÓN-MONTELONGO, I. S.; NIETO-VILLAR, J. M. The rössler system as a model for chronotherapy. *Journal of Physics: Conference Series*, vol. 23, no. 1, p. 58, jan 2005. Available from Internet: <https://dx.doi.org/10.1088/1742-6596/23/1/006>.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. (Information science and statistics).
- BRENNAN, M.; KOVACIC, I. Examples of physical systems described by the duffing equation. In: \_\_\_\_\_. *The Duffing Equation: Nonlinear Oscillators and Their Behaviour*. [S.l.]: Wiley, 2011.
- BRUNTON, S.; KUTZ, J. N.; PROCTOR, J. L. Data-driven discovery of governing physical laws. *SIAM NEWS*, vol. 50, no. 01, 2017.
- BRUNTON, S. L.; PROCTOR, J. L.; KUTZ, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, p. 3932–3937, 2016.
- BRUNTON, S. L. et al. Discovery of physics from data: Universal laws and discrepancy models. *CoRR*, abs/1906.07906, 2019.
- CAMASTRA, F.; VERRI, A. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, p. 801–805, 2005.
- CHEN, S. S.; DONOHO, D. L.; SAUNDERS, M. A. Atomic decomposition by basis pursuit. *SIAM Review*, vol. 43, no. 1, p. 129 – 159, 2001.
- CORBETTA, M. Application of sparse identification of nonlinear dynamics for physics-informed learning. In: *2020 IEEE Aerospace Conference*. [S.l.: s.l.], 2020. p. 1–8.
- CORTIELLA, A.; PARK, K.-C.; DOOSTAN, A. A Priori Denoising Strategies for Sparse Identification of Nonlinear Dynamical Systems: A Comparative Study. *Journal of Computing and Information Science in Engineering*, vol. 23, no. 1, p. 011004, 07 2022. ISSN 1530-9827. Available from Internet: <https://doi.org/10.1115/1.4054573>.
- COTTONE, F.; VOCCA, H.; GAMMAITONI, L. Nonlinear energy harvesting. *Physical Review Letters*, American Physical Society, vol. 102, p. 080601, 2009.

- COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, vol. 13, no. 1, p. 21–27, 1967.
- Cunha Jr, A.; LOPES, D. M. S. *Inference of nonlinear dynamical systems from data using sparse regression*. 2021.
- DANTAS, E.; TOSIN, M.; Cunha Jr, A. Calibration of a SEIR-SEI epidemic model to describe the Zika virus outbreak in Brazil. *Applied Mathematics and Computation*, vol. 338, p. 249–259, 2018.
- DANTAS, E.; TOSIN, M.; Cunha Jr, A. *An uncertainty quantification framework for a Zika virus epidemic model*. 2019. Conference of Computational Interdisciplinary Science 2019.
- EAGLE, D. *A MATLAB Implementation of the BV78 ODE Solver*. 2023. (<https://www.mathworks.com/matlabcentral/fileexchange/55014-a-matlab-implementation-of-the-bv78-ode-solver>). MATLAB Central File Exchange. Access on: November 7, 2023.
- EFRON, B. et al. Least angle regression. *The Annals of Statistics*, Institute of Mathematical Statistics, vol. 32, no. 2, p. 407 – 499, 2004.
- FUKAMI, K. et al. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics*, Cambridge University Press, vol. 926, p. A10, 2021.
- GLASER, J. I. et al. The roles os supervised machine learning in systems neuroscience. *Progress in Neurobiology*, Elsevier, vol. 175, p. 126–137, 2019.
- GRUBER, M. *Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators*. [S.l.]: Taylor & Francis, 1998. (Statistics: A Series of Textbooks and Monographs).
- GUYOMAR, D. et al. Energy harvesting using non-linear techniques. In: \_\_\_\_\_. *Energy harvesting technologies*. Boston, MA: Springer, 2009.
- Géron, A. *hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2. ed. [S.l.]: O'Reilly Media, 2019.
- HANS, C. Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, Taylor Francis, vol. 106, no. 496, p. 1383–1393, 2011.
- HASSANIBESHELI, F.; BOERS, N.; KURTHS, J. Reconstructing complex system dynamics from time series: a method comparison. *New Journal of Physics*, 2020.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY: Springer, 2009.
- HE, G.; ZHAO, Y.; YAN, C. Mflp-pinn: A physics-informed neural network for multiaxial fatigue life prediction. *European Journal of Mechanics - A/Solids*, vol. 98, p. 104889, 2023. ISSN 0997-7538.
- HERBRICH, R. *Learning Kernel Classifiers*. [S.l.]: MIT Press, 2002. (Adaptive computation and machine learning).

- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, vol. 9, no. 8, p. 1735–1780, 1997.
- HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, Taylor Francis, vol. 12, no. 1, p. 55–67, 1970.
- HONIGBAUM, J.; ROCHINHA, F. A. Data-driven identification of coupling closure equations in vortex-induced vibrations phenomenological models. *Ocean Engineering*, vol. 266, p. 112981, 2022. ISSN 0029-8018.
- JACOBS, M. et al. Hypersindy: Deep generative modeling of nonlinear stochastic governing equations. *arXiv preprint arXiv:2310.04832*, 2023.
- JR, A. C.; BARTON, D. A.; RITTO, T. G. Uncertainty quantification in mechanistic epidemic models via cross-entropy approximate bayesian computation. *Nonlinear Dynamics*, Springer, vol. 111, no. 10, p. 9649–9679, 2023.
- KENNEDY, M.; CHUA, L. Van der pol and chaos. *IEEE Transactions on Circuits and Systems*, vol. 33, no. 10, p. 974–980, 1986.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, vol. 521, p. 436–444, 2015.
- LI, H. et al. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, vol. 45, p. 17–26, 2014. ISSN 0968-090X. Advances in Computing and Communications and their Impact on Transportation Science and Technologies.
- LIU, R. et al. Principal component regression analysis with spss. *Computer Methods and Programs in Biomedicine*, vol. 71, no. 2, p. 141–147, 2003. ISSN 0169-2607.
- LOPES, D.; Cunha Jr, A. On the physical consistency of evolution laws obtained with sparse regression. In: LACARBONARA, W. et al. (Ed.). *Advances in Nonlinear Dynamics*. Cham: Springer International Publishing, 2022. p. 463–473. ISBN 978-3-030-81166-2.
- LOPES, D. M. S.; Cunha Jr, A. A data-driven approach for inference of the evolution equation of a duffing oscillator. In: . [S.l.: s.l.], 2019.
- LOPES, V.; PETERSON, J.; Cunha Jr, A. *Numerical study of parameters influence over the dynamics of a piezo-magneto-elastic energy harvesting device*. 2017. CNMAC 2017 XXXVII Congresso Nacional de Matemática Aplicada e Computacional.
- LYDON, B.; POLAGYE, B.; BRUNTON, S. Nonlinear wec modeling using sparse identification of nonlinear dynamics (sindy). In: *Proceedings of the European Wave and Tidal Energy Conference*. [S.l.: s.l.], 2023. vol. 15.
- MARBLESTONE, A. H.; WAYNE, G.; KORDING, K. P. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, vol. 10, p. 94, 2016.
- MATOS, D.; JR, A. C. Inference of nonlinear dynamical systems from data using sparse regression. In: *14th World Conference on Computational Mechanics*. [S.l.: s.l.], 2021.

- MESSENGER, D. A.; BORTZ, D. M. Weak sindy for partial differential equations. *Journal of Computational Physics*, vol. 443, p. 110525, 2021. ISSN 0021-9991. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0021999121004204>.
- MILLS, K.; SPANNER, M.; TAMBLYN, I. Deep learning and the Schrödinger equation. *Physical Review A*, American Physical Society (APS), vol. 96, no. 4, 2017.
- NATH, K. et al. Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines. *arXiv preprint arXiv:2304.13799*, 2023.
- NOCEDAL, J.; WRIGHT, S. *Numerical Optimization*. New York, NY: Springer, 2006.
- NORENBERG, J. P. et al. Probabilistic maps on bistable vibration energy harvesters. *arXiv preprint arXiv:2302.12769*, 2023.
- ODEN, J. T. *An introduction to mathematical modeling: a course in mechanics*. [S.l.]: John Wiley & Sons, 2011.
- PANCHAL, J. H. et al. Special Issue: Machine Learning for Engineering Design. *Journal of Mechanical Design*, vol. 141, no. 11, 10 2019. ISSN 1050-0472.
- PATHAK, J. et al. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, vol. 120, p. 024102, 2018.
- PETERSON, J.; LOPES, V.; Cunha Jr, A. *Maximization of the electrical power generated by a piezo-magneto-elastic energy harvesting device*. 2016. XXXVI Congresso Nacional de Matemática Aplicada e Computacional.
- POL, B. van der. Lxxxviii. on “relaxation-oscillations”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor Francis, vol. 2, no. 11, p. 978–992, 1926.
- RICHARDS, B.; LILLICRAP, T.; BEAUDOIN, P. e. a. A deep learning framework for neuroscience. *Nature*, vol. 22, p. 1761–1770, 2019.
- RITTO, T. G.; JR, A. C.; BARTON, D. A. Parameter calibration and uncertainty quantification in an seir-type covid-19 model using approximate bayesian computation. In: *3rd Pan American congress on computational mechanics (PANACM 2021)*. [S.l.: s.l.], 2021.
- ROCA, L. De la et al. *Control of chaos via OGY method on a bistable energy harvester*. Uberlândia-MG, Brazil: [s.n.], 2019. 25th ABCM International Congress on Mechanical Engineering.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, vol. 65, no. 6, 1958.
- RYAN, T. P. *Modern regression methods*. 2. ed. [S.l.]: Wiley-Interscience, 2008.
- RÖSSLER, O. An equation for continuous chaos. *Physics Letters A*, vol. 57, no. 5, p. 397–398, 1976. ISSN 0375-9601.

- SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, vol. 3, p. 210–229, 1959.
- SCHWAB, K. *The Fourth Industrial Revolution*. [S.l.]: The Fourth Industrial Revolution, 2016.
- SMOLA, A.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and Computing*, vol. 14, p. 199–222, 2004.
- TIBSHIRANI, R. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, Royal Statistical Society, Wiley, vol. 58, no. 1, p. 267–288, 1996.
- VAMVOUDAKIS, K. G. et al. Autonomy and machine intelligence in complex systems: A tutorial. In: *2015 American Control Conference (ACC)*. [S.l.: s.l.], 2015. p. 5062–5079.
- VILLANI, L.; SILVA, S.; Cunha Jr, A. Damage detection in uncertain nonlinear systems based on stochastic Volterra series. *Mechanical Systems and Signal Processing*, vol. 125, p. 288–310, 2018.
- VILLANI, L. et al. Damage detection in an uncertain nonlinear beam based on stochastic Volterra series: An experimental application. *Mechanical Systems and Signal Processing*, vol. 128, p. 463–478, 2019.
- WENTZ, J.; DOOSTAN, A. Derivative-based sindy (dsindy): Addressing the challenge of discovering governing equations from noisy data. *Computer Methods in Applied Mechanics and Engineering*, vol. 413, p. 116096, 2023. ISSN 0045-7825. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0045782523002207>.
- XU, M.; DAVID, J. M.; KIM, S. H. The fourth industrial revolution: opportunities and challenges. *International journal of financial research*, vol. 9, no. 2, 2018.
- YANG, Y.; MOHAMED, A. B.; PERDIKARIS, P. Bayesian differential programming for robust systems identification under uncertainty. *Proceedings of the Royal Society A*, vol. 476, no. 20200290, 2020.
- YANO, M. O. et al. Damage quantification using transfer component analysis combined with gaussian process regression. *Structural Health Monitoring*, vol. 22, no. 2, p. 1290–1307, 2023.
- ZDUNIAK, B.; BODNAR, M.; FORYŚ, U. A modified van der pol equation with delay in a description of the heart action. *International Journal of Applied Mathematics and Computer Science*, vol. 24, no. 4, p. 853–863, 2014. Available from Internet: <https://doi.org/10.2478/amcs-2014-0063>.
- ZENG, S. et al. Optimized sparse polynomial chaos expansion with entropy regularization. *Adv. Aerodyn*, vol. 4, no. 3, 2022.
- ZHANG, D.; GUO, L.; KARNIADAKIS, G. E. Learning in modal space: Solving time-dependent stochastic PDEs using physics-informed neural networks. *CoRR*, abs/1905.01205, 2019.

ZHANG, Z. et al. Dynamic reliability analysis of nonlinear structures using a duffing-system-based equivalent nonlinear system method. *International Journal of Approximate Reasoning*, vol. 126, p. 84–97, 2020. ISSN 0888-613X.

ZHANG, Z. et al. Discovering a reaction–diffusion model for alzheimer’s disease by combining pinns with symbolic regression. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, vol. 419, p. 116647, 2024.

ZHENG, P. et al. A unified framework for sparse relaxed regularized regression: SR3. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), vol. 7, p. 1404–1423, 2019.

ZIO, S.; ROCHINHA, F. A. Data-driven calibration of p3d hydraulic fracturing models. *International Journal for Uncertainty Quantification*, Begel House Inc., vol. 10, no. 4, 2020.