Automatic Extraction of Ordinary Differential Equations from Data: Sparse Regression Tools for System Identification

EGAN, KEVIN

Automatic Extraction of Ordinary Differential Equations from Data:

Sparse Regression Tools for System Identification

Kevin Egan

A Thesis presented for the degree of
Doctor of Philosophy

Department of Engineering
Durham University
United Kingdom
March 2023

To Mom, Dad, Kelsey, and Brendan
Thank you for your unconditional support through thick and thin.

Abstract

Studying nonlinear systems across engineering, physics, economics, biology, and chemistry often hinges upon successfully discovering their underlying dynamics. However, despite the abundance of data in today's world, a complete comprehension of these governing equations often remains elusive, posing a significant challenge. Traditional system identification methods for building mathematical models to describe these dynamics can be time-consuming, error-prone, and limited by data availability. This thesis presents three comprehensive strategies to address these challenges and automate model discovery. The procedures outlined here employ classic statistical and machine learning methods, such as signal filtering, sparse regression, bootstrap sampling, Bayesian inference, and unsupervised learning algorithms, to capture complex and nonlinear relationships in data. Building on these foundational techniques, the proposed processes offer a reliable and efficient approach to identifying models of ordinary differential equations from data, differing from and complementing existing frameworks. The results presented here provide rigorous benchmarking against state-of-the-art algorithms, demonstrating the proposed methods' effectiveness in model discovery and highlighting the potential for discovering governing equations across applications such as weather forecasting, chemical reaction and electrical circuit modelling, and predator-prey dynamics. These methods can aid in solving critical decision-making problems, including optimising resource allocation, predicting system failures, and facilitating adaptive control in various domains. Ultimately, the strategies developed in this thesis are designed to integrate seamlessly into current workflows, thereby promoting data-driven decision-making and enhancing understanding of complex system dynamics.

Declaration

The work in this thesis is based on research carried out at the Department of Engineering, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Parts of this work will be submitted to the following refereed journals. The following provides further detail regarding each author's contribution to the papers.

- Chapter 3
  - I have submitted Chapter 3 to Communications Physics with Weizhen Li and Rui Carvalho and am awaiting revisions.
  - Weizhen Li collaborated on ideas, provided significant insight and suggested the theory behind the systematic analysis in this paper's results section. Weizhen also created the schematic, helped develop the frequency distribution plots, and helped analyse the results. Without Weizhen's exhaustive efforts, the paper would not have the strong results displayed within this thesis.
  - Dr Rui Carvalho provided expansive insight, ideas, and collaboration for writing the paper. Rui helped develop the method, along with editing, maintaining, and aligning the paper so that it was written in the style

of the top journals. Rui's constant time, leadership, and guidance have significantly impacted the structure and overall quality of the paper.

- Chapter 4
  - I plan to submit Chapter 4 to IEEE Access upon receiving feedback for Chapter 3 as similar methods are used for algorithm evaluation, and I aim to avoid crossover.
  - Weizhen Li helped with the initial proposal of Otsu's method for thresholding.
  - Dr Rui Carvalho provided oversight and suggestions for improvement on the text.
- Chapter 5
  - Upon receiving feedback from Communications Physics for Chapter 3, I plan to submit Chapter 5 to Scientific Reports or Communications Physics as similar methods are used in both chapters, and I aim to avoid crossover.
  - Dr Rui Carvalho provided edits and insight for writing the paper.

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to Dr Rui Carvalho. As my supervisor, Rui consistently challenged me to better myself, demonstrating remarkable patience and guidance in helping me achieve my goals throughout this research project. Rui's insights significantly deepened my understanding of each method discussed in this thesis, imparting the invaluable lesson of diligent exploration and leaving no stone unturned. I have been honoured to learn from Rui's rigorous and meticulous approach to problem-solving and truth-seeking, a mentality I intend to uphold throughout my career and life.

I also extend my heartfelt thanks to my second supervisor Dr Hongjian Sun for his guidance during my research project. Hong has provided me with opportunities, support, and secondary advice for a fruitful career in machine learning, and his positive attitude has made a lasting impression on my personal life.

Thanks also go to Dr Behzad Kazemtabrizi and Dr Stefano Giani, who provided me with significant advice throughout my progression reviews, which helped me better understand the project and how to explain my findings throughout this work better. Furthermore, I would like to thank the Department of Engineering at Durham University, which provided me with the funding to perform my research through the Durham Doctoral Studentship programme.

I want to thank Weizhen Li, who helped me develop plots, gave me insight into my writing, and pushed me to become a better data scientist and statistician. Spending time in the office with Weizhen discussing different fields of machine learning and system identification and competing over whose code was better for implementation has given me some of the best experiences in my life. I appreciate the countless hours and help he has provided for me to accomplish this project.

I sincerely appreciate my closest friends in Durham who have tolerated my relentless antics and eccentricities. Thanks to Arnau and Carol for never relenting in remembering to laugh and enjoy our time together and also for enabling me to use their claims as my trump card to support my assertions in debates. From tedious arguments with Alan, discussing radical social change and digestive well-being

with Jen, post-work walks with Orlagh, to debating the definition of science with Pietro, each of these individuals has lent their strength and support during my most challenging times. The many nights discussing the definitions and metaphors for precision and recall, the bias-variance trade-off, and XGBoost with Sean and James have helped me gain confidence and improve as a data scientist, all while the rest of the group begged us to save the machine learning infested conversations at work. I could not have achieved this feat without these people and the support they have given me throughout the years.

Thanks to Emily, who inspired me to journey across the pond and achieve this ambitious goal. As my undergraduate supervisor, Dr Andrews, warned, this project was in fact a slough, demanding every ounce of my energy. I appreciate all your support during my time in Durham and the days spent at the pub getting away from work. Without you, I would not have the surprisingly extensive knowledge of England I carry with me today.

Lastly, I would like to thank my siblings and parents. Their constant support has helped me stay positive and work hard throughout my project, and I cannot express my thanks enough for their belief in my success.

Contents

List of Figures

List of Tables

List of Symbols

General Notation

$\mathbb{R}$	Real numbers, p. 7
$\mathbb{E}$	Expected value of a function, p. 14
n	Number of observations, p. 7
t	Continuous time measurements, p. 45
p	Number of predictors, p. 7
m	Number of state space variables, p. 45
ϵ	Irreducible error from noise, p. 9
x_{ij}	The ith observation of the jth predictor, p. 7
$x(t)$	State of the system at times $t_1, t_2, \ldots, t_n$, p. 45
$\mathbf{X}$	Data matrix ($N \times p$) in Chapter 1 & the state space matrix ($N \times m$) for subsequent chapters, p. 7
$\dot{\mathbf{X}}$	Derivative matrix of $\mathbf{X}$ ($N \times m$), p. 46
$\mathbf{y}$	Dependent variable vector, (length n), p. 9
$\hat{\mathbf{y}}$	Estimated values of $\mathbf{y}$, (length n), p. 9
$\Theta(\mathbf{X})$	Design matrix for system identification ($N \times p$), p. 46
$\theta(x)_{\mathrm{F}}$	Feature vector of symbolic functions ($p \times 1$), p. 46
β	Vector of regression coefficients ($p \times 1$), p. 11

$\mathbf{B}$	Matrix containing β for each state space variable ($p \times m$), p. 46
w	Regression weights vector (px1), p. 31
$\mathbf{E}$	Gaussian random noise matrix distributed with zero mean and standard deviation one (Nxp), p. 46
B	Number of bootstrap samples, p. 37
CI_{lo}	Lower confidence interval bound, p. 38
CI_{up}	Upper confidence interval bound, p. 38
α	Significance level for confidence bound, p. 38
loglik	Log-likelihood function, p. 25
λ	Sparse regression tuning parameter, p. 27
$\widehat{M}_n$	Statistical learning algorithm for random regression function, p. 17

Nomenclature

RSS	Residual sum of squares, p. 11
TSS	Total sum of squares, p. 13
OLS	Ordinary least squares, p. 11
MSE	Mean squared error, p. 12
RMSE	Root mean squared error, p. 13
AIC	Akaike information criterion, p. 55
BIC	Bayesian information criterion, p. 33
VIF	Variance inflation factor, p. 24
IID	Independent and identically distributed terms, p. 25
PMU	Phasor Measurement Units, p. 109
MLE	Maximum likelihood estimation, p. 26
SNR	Signal-to-noise ratio, p. 34
MCMC	Markov chain Monte Carlo, p. 39
ODE	Ordinary differential equation, p. 45
PDE	Partial differential equation, p. 45
FFT	Fast Fourier Transform, p. 110

lasso	Least absolute shrinkage and selection operator, p. 27
ARGOS	Automatic regression for governing equations, p. 5
SINDy	Sparse identification of nonlinear dynamics, p. 53
ASINDy	Adaptive sparse identification of nonlinear dynamics, p. 5
ARGOS-BI	Automatic regression for governing equations with Bayesian Inference, p. 5

CHAPTER 1

Introduction

1.1 Background

In the modern era of data-centric engineering, machine learning and data science are improving practices across various domains, such as materials science [1–4], manufacturing [5–7], operations [8, 9], control [10–15], and construction [16, 17]. Engineers employ machine learning algorithms to build prediction models from empirical data that enhance the dependability, safety, and efficiency of real-world systems. Moreover, these mathematical models are pivotal in describing the governing dynamics of natural phenomena [18]. From studying turbulent flow in aeronautics [19–21], exploring the connectivity of the brain in biological engineering [22–24], and explaining the actions of financial markets [25–27], machine learning is driving advancements in many diverse areas. Furthermore, its significant impact has drawn the attention of leading research institutions worldwide.

Led by forward-thinking individuals, The Alan Turing Institute in the United Kingdom has established research programmes to harness the power of machine learning and data science and revolutionise data-centric engineering [28]. The versatility of these methods is evidenced by their application in smart city develop-

ment [29–31], where they have been used to optimise urban mobility [32, 33], energy consumption [34–36], waste management [37–39], and air quality [40–42].

One process gaining popularity within data-centric engineering is system identification. This crucial framework focuses on deriving governing equations through an inverse problem procedure [43, 44]. By directly examining observational data, engineers leverage statistical theory to assess the predictive capabilities of several potential models. After rigorous evaluation, these engineers can determine the optimal mathematical representation of the underlying system [45]. Machine learning has been instrumental in enhancing system identification processes, as it facilitates the determination of symbolic terms that describe complex dynamics from data [46–49].

Integrating statistical inference with machine learning algorithms has led to a new paradigm in the field, prompting scientists and engineers to focus on assessing and developing more optimal prediction models. These tools have demonstrated the ability to learn the underlying patterns and relationships in large data sets, examining observations generated by a system and recognising their defining features [50–52]. Furthermore, these methods offer a rigorous approach to model selection and parameter estimation, reducing potential biases in the identification process [44].

Despite significant progress in identifying differential equations with symbolic regression [53–56] and probabilistic methods [48, 57–59], as well as optimising predictions with deep learning [7, 60, 61] and compressive sensing [62], challenges in interpreting these models persist [63]. However, as the field of system identification advances, sparse regression has shown promise, empowering engineers to perform variable selection and identify the governing equations that describe the behaviour of complex dynamics [46, 64–66]. This approach, grounded in statistics, has the potential to lead to faster and more accurate discoveries across various disciplines, incentivising scientists and engineers to evaluate their models more carefully and ensure the assumptions of these techniques and their predictions align with the observed data. Furthermore, these methods can lead to the automation of the model discovery process, which can enhance efficiency, scalability, and reproducibility and facilitate the optimisation of the parameters of a given dynamical system [18].

This thesis aims to contribute to the growing trend of automating the discovery

of underlying equations governing system dynamics from data by combining sparse regression with bootstrap sampling, Bayesian methods, and clustering algorithms. Unlike previous strategies that have implemented sparse regression [46, 64, 67, 68], the methods here enable engineers to effectively adapt to different forms of ordinary differential equations without requiring any manual tuning parameters. Furthermore, the inference-based processes proposed here advance the field of system identification by solving the inverse problem reliably with classic model assessment methods. Ultimately, these developments will hopefully shape the future of scientific methods for automatically discovering elusive laws describing many intricate systems.

1.2 Motivation

Throughout my academic career, I have cultivated a fervent interest in statistical and machine learning applications, consistently exploring avenues to harness these tools for societal betterment. My journey began with the intention to model the spatial and temporal usage behaviour of electric vehicle (EV) charging stations. Reliable models in this domain promise to aid cities in strategically placing new stations, reducing user waiting times, and promoting electric vehicle adoption—leading to decreased greenhouse gas emissions.

However, I faced two critical challenges: limited data and, importantly, a need for a novel, automated approach to system identification. The availability of comprehensive, high-quality data on EV charging station usage patterns was restricted, which posed difficulties in developing predictive models that were both accurate and reliable. Additionally, I quickly realised that even if I could obtain such data, the behaviour of EV users is governed by a myriad of factors, both predictable (like battery capacity or charging station distribution) and unpredictable (like user preference or unforeseen events). This complexity led me to an important insight: instead of directly predicting the behaviour, why not first identify the underlying dynamical system governing it?

This pivot in research direction opened up the field to even more opportunities.

Dynamical systems were not confined to understanding EV charging stations but spanned various critical domains such as drug discovery, autonomous machines, epidemiology, and more. My refined research motivation thus became clear: to unearth, understand, and automate the discovery of these governing dynamical systems that dictate behaviours across diverse fields. The broader implications were thrilling: an automated, reliable method to uncover these systems could revolutionise how various sectors approach challenges that involve better understanding processes within data.

Traditional system identification relies heavily on domain-specific expertise, often making it labour-intensive and less generalisable. In contrast, my thesis aspires to contribute to the automation of this process, harnessing the power of machine learning and statistics. This endeavour is not just about the EVs but about providing a toolkit to the scientific community to better understand and predict phenomena in fields as diverse as physics, chemistry, biology, neuroscience, and environmental science.

We can unlock the potential to predict behaviours across many real-world systems by elucidating and offering algorithms that can automatically deduce governing equations. Such advancements are academically fascinating and can drive practical benefits, especially when data is scarce or hard to obtain. Therefore, my work's vision is dual-layered: firstly, to drive advancements in system identification through machine learning and to motivate researchers from various disciplines to employ these innovations, refining their methodologies and fostering enhanced discoveries.

1.3 Thesis Structure

This dissertation presents the development of innovative computational machine learning tools for automating the discovery of dynamical systems from data. The structure of this work addresses a series of fundamental research questions, each leading to the next, ensuring clarity of purpose and highlighting the novelty of the proposed methodologies.

1. **Chapter 2 (Literature Review):**

What existing statistical methods form the foundation for automating the discovery of dynamical systems?

This chapter provides a comprehensive review of statistical and machine learning methods, emphasising regression methodologies, unsupervised learning algorithms, and system identification techniques, all as the groundwork for the research presented herein.

2. **Chapter 3 (Automatic Regression for Governing Equations (ARGOS)):**

   How can model discovery of ordinary differential equations be advanced by integrating sparse regression with statistical inference?

   A novel frequentist approach for advancing the field with sparse regression is introduced, demonstrating its superiority over state-of-the-art implementations for model discovery.

3. **Chapter 4 (Adaptive SINDy (ASINDy)):**

   How can clustering-based algorithms refine and enhance the sparse identification of nonlinear dynamics?

   The chapter presents Adaptive-SINDy (ASINDy), an extension of the sparse identification of nonlinear dynamics approach that harnesses clustering methodologies to automatically discern the optimal thresholding parameter, showcasing discernible enhancements over traditional mechanisms.

4. **Chapter 5 (Automatic regression for governing equations with Bayesian Inference (ARGOS-BI)):**

   How can integrating Bayesian methods into the ARGOS methodology lead to improved system identification?

   With a Bayesian perspective, this chapter augments the ARGOS method, yielding robust results against noise, optimising efficiency, and demonstrating successful system identification with fewer data samples.

5. **Chapter 6 (Conclusion):**

How has this thesis advanced the field of system identification, and what avenues have been highlighted for upcoming investigations?

This chapter amalgamates the pioneering methodologies propounded, illuminating the strides in data-driven system identification and paving avenues for ensuing research endeavours.

After establishing the foundational principles in the literature review, Chapter 3 introduces ARGOS. The chapter sheds light on ARGOS' enhanced performance in system identification through sparse regression, focusing on its resistance to noise and the employment of bootstrap sampling for effective variable selection from temporal data sets.

Chapter 4 outlines the Adaptive-SINDy (ASINDy) method, an innovative clustering-based approach to enhance sparse identification. It offers an inventive process for determining the optimal sparsity-promoting parameters, showcasing its efficacy over traditional mechanisms.

In Chapter 5, the ARGOS approach is reimagined through a Bayesian lens. Unlike the frequentist bootstrap sampling approach, Bayesian regression is applied to determine the optimal prediction model from credible posterior intervals. This method provides the added benefit of increasing efficiency while displaying more noise-robust results and requiring fewer observations to identify the underlying system. Moreover, the chapter develops a comparison with a similar extension of the sparse identification of nonlinear dynamics, which implements an ensembling procedure.

Central to this dissertation is a novel systematic analysis dedicated to evaluating the efficacy of each algorithm in automatically discovering nonlinear systems from data. The process expands ordinary differential equations using random initial conditions, thereby allowing for a more general representation of the ability of each algorithm rather than studying the results of a method for one data set. Two distinct test sets have been instituted to ascertain the data quality and quantity essential for optimising each algorithm's performance, focusing on increasing the number of observations and modulating the signal-to-noise ratio.

The concluding chapter summarises the methodologies discussed, illuminating

the advances in data-driven system identification. The culmination of this research offers the engineering community new horizons for data exploration, embodying significant contributions to automating model discovery.

1.4 Notation

The notation in this report is similar to [50, p.10; 51,52]. As such, n represents the number of distinct data points or observations in the sample distribution, while p represents the number of variables for prediction, sometimes referred to as predictors. Lowercase bold will always denote a vector of length n:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix},$$

while lowercase normal font, a, describes vectors not of length n. Here, the lowercase normal font also defines scalars, a. If there is a situation in which these two cases are challenging to interpret, clarity will be provided for the intended use. Bold capitals denote label matrices, e.g., $\mathbf{A}$. Regardless of their dimensions, the normal capital font will designate random variables, e.g., A, and specify whether an object is an $r \times s$ matrix with $\mathbf{A} \in \mathbb{R}^{r \times s}$ [50, p.11].

The ith observation of the jth predictor is denoted as x_{ij}, such that $i = 1, 2, ..., n$ and $j = 1, 2, ..., p$. The description of matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ provides x_{ij} is its (i, j)th element. Furthermore, the matrix $\mathbf{X}$ contains n rows and p columns:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

The rows of $\mathbf{X}$ as $x_1, x_2, ..., x_n$ denote x_i as a vector with length p that stores p predictors for the ith observation [50, p.10; 51,52]. Here, vectors are typically

represented as columns

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}.$$

When describing the columns of $\mathbf{X}$ as $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p$, vectors are of length n,

$$\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}.$$

The matrix $\mathbf{X}$ can then appear as

$$\mathbf{X} = (\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_p),$$

or

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}.$$

The transpose of a matrix or vector is

$$\mathbf{X}^T = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \cdots & x_{np} \end{pmatrix},$$

and

$$x_i^T = (x_{i1} \ x_{i2} \ \cdots \ x_{ip}).$$

Furthermore, y_i denotes the ith observation of the dependent variable and represents

the set of all n observations in vector form as

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

With this notation, the observed data takes the form of $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. When making predictions or estimating unknown parameters, the predicted values are represented using "*hat*" notation, e.g., $\hat{\mathbf{y}}$ is the vector of predicted values for $\mathbf{y}$ [50, p.11; 51,52].

Training data allows the construction of subsets from the original data set. Measurements for the training data are defined as (x_i, y_i), $i = 1, \ldots, n$, which helps develop prediction models throughout this thesis [51, p.11]. The training set approach allows for model evaluation since observations are withheld from the training data set to assess the accuracy of the statistical learning model. The application of this approach can vary depending on the type of data. For instance, in a time-series setting, statistical learning models train on past observations, and scientists and engineers assess the model's predictive performance using future observations found in the remaining test data. Later sections provide a more in-depth discussion of this approach.

Finally, X denotes the input or *independent* variables, and the subscript X_j distinguishes them. Similarly, Y represents the output or the *dependent* variable [50, p.15; 51,52]. Machine learning methods assume that there is a relationship between Y and $X = (X_1, X_2, \ldots, X_p)$, whose description takes the form

$$Y = f(X) + \epsilon, \tag{1.1}$$

where f is some fixed but unknown function of $X_1, \ldots, X_p$, and ϵ is a random *error term*, which is independent of X and has mean zero [50, p.16]. In Eq. (1.1), f develops a structured representation of Y with the information from X.

CHAPTER 2

Literature Review

While data continues to be collected, stored, and manipulated in prolific quantities, the ability to automatically extract governing equations from this information remains arduous and elusive. In the face of this challenge, system identification, the process of building mathematical models based on observational data, has gained attention. Specifically, sparse modelling has become increasingly popular due to its ability to uncover a parsimonious and interpretable representation of the data's governing equations by seeking models with only a small number of nonzero parameters. By applying this approach, engineers have leveraged a sparse methodology across various domains, including signal processing, control, machine learning, and neuroscience. This chapter introduces the fundamental concepts of sparse modelling for system identification, including methods for selecting the most relevant features, estimating the model parameters, and model evaluation techniques applied throughout this thesis.

2.1 Model Accuracy

Model accuracy defines a statistical learning method's ability to develop robust predictions for the data. This concept is special in regression analysis, where establishing a relationship between dependent and independent variables is key. This thesis employs linear regression, an essential method for developing interpretable models, with further details in Section 2.2.1. The essence of model accuracy hinges on its congruence with actual observations, underscoring the need to minimise prediction errors. Consequently, this section outlines the fundamental tools for assessing accuracy in statistical learning and system identification.

2.1.1 Measuring the Quality of Fit

Residual Sum of Squares

To begin, the notation $\hat{y}_i = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i x_i$ provides the prediction for Y based on the ith value of X. Subsequently, $e_i = y_i - \hat{y}_i$ determines the ith residual, representing the difference between the observed value and the predicted estimate given by a regression model [50, p.61]. Moreover, the most common error metric in regression is the residual sum of squares (RSS)

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \end{aligned} \tag{2.1}$$

where $\hat{f}(x_i) = \hat{y}_i$ is the prediction of the ith observation of $\hat{\mathbf{y}}$ [50, p.29]. Equation (2.1) demonstrates the deviation between the predicted values of the model and the observed data and thus explains how well the regression model fits the data set.

Ordinary least squares (OLS) regression predicts Y by calculating the optimal values of $\hat{\beta}_0$ through $\hat{\beta}_p$ to minimise the RSS. Thus, the optimal regression model fit provides predicted values closest to the data set's actual observations. Since Eq. (2.1) is squared and returns a nonnegative quantity, the best model is the one

whose RSS is closest to zero. In contrast, a higher RSS indicates that the model fits the data poorly.

Mean Squared Error

The mean squared error (MSE) provides an alternative measure for model accuracy by calculating the average squared difference between the predicted values of the model and the actual measurements of the data set. It corresponds to the expected value of the squared error loss and is defined as [50, p.29]

$$\text{MSE} = \frac{1}{n} \cdot \text{RSS}. \tag{2.2}$$

A regression model reduces the MSE by selecting predicted responses close to the observations in the data set. In contrast, when the MSE is large, the predicted responses and the actual measurements differ substantially.

Typically, the training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ helps determine the MSE for a given prediction model. These observations provide an estimate $\hat{f}$ and help calculate $\hat{f}(x_1), \hat{f}(x_2), \ldots, \hat{f}(x_n)$. However, in the time-series context, the training MSE only informs researchers of their model's performance on past observations, and in general, they are less interested in whether $\hat{f}(x_i) \approx y_i$. Rather, the fit of the model $\hat{f}(x_0)$ to a given point in the test data (x_0, y_0) is often far more desired because it is data that the statistical learning model has not previously assessed [50, p.30]. Therefore, assuming there is a large enough test data set, the resulting metric takes the form

$$\text{Ave}(y_0 - \hat{f}(x_0))^2. \tag{2.3}$$

Here, Eq. (2.3) describes the average prediction error for the test observations. Thus, the smallest test MSE determines the optimal prediction model. If a model results in a small training MSE but a large test MSE, it overfits the training data and may not be the best prediction model for the test data. For example, statistical learning models overfit data sets when they detect random patterns in the training data that may not be properties of the unknown function [50, p.32]. Under this assumption, the training MSE is not a sufficient metric for model deployment and sufficient test

data is required to determine a reliable model.

The root mean squared error (RMSE) provides an alternatively relevant metric by taking the square root of the MSE. In this case, measuring the standard deviation of an observed value allows scientists and engineers to evaluate the accuracy of a given method, clarifying the ability of the model to make predictions. This approach adds the effect of giving relatively high weight to significant errors and provides a more interpretable representation of the error, placing it on the same scale as the dependent variable.

R^2 Statistic

The R^2 statistic, also known as the coefficient of determination, determines the proportion of variability in Y explained by X [50, p.69]. This statistical measure results in a value between 0 and 1 and is independent of the scale of Y. The R^2 metric is calculated as:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}, \tag{2.4}$$

where $\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ denotes the total sum of squares (TSS), quantifying the total variance in Y. Here, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ denotes the mean of all n observed values of the dependent variable Y. From Eq. (2.1), the RSS describes the variability left unexplained after performing regression. Alternatively, the TSS measures the total variance or variability in Y before performing regression. Therefore, measuring $\text{TSS} - \text{RSS}$ determines how well a given model explains the variability in Y. Hence, the R^2 statistic represents the proportion of variability in Y that X can explain [50, p.70]. Ultimately, a regression model with an R^2 statistic closer to 1 better explains the variability in the response.

2.1.2 The Bias-Variance Trade-Off

Statistical and machine learning algorithms are essential tools that aim to make predictions based on complex patterns in real-world observations. However, their models often misrepresent the predictor variable due to noise and dependence be-

tween variables found in data sets [69, p.63]. Dense, complex models may better fit the training data by containing more variables but often suffer from overfitting and may not generalise well to the test data [50, p.22; 70, p.145]. In contrast, simpler models may only capture some of the nuances of the data but can be more interpretable and make more accurate predictions on new data. Striking a balance between model complexity and prediction accuracy is critical, and this trade-off is commonly addressed by considering bias and variance. The following section explores how understanding the bias-variance trade-off can help improve the ability of statistical and machine learning models to make better predictions.

Expected Value of the Dependent Variable

To understand the bias-variance trade-off completely, it is imperative to define several important terms, the first being the expected value of the dependent variable. Given $x \in \mathbb{R}^{p \times n}$, the expected value of the dependent variable is denoted as

$$f(x) = \mathbb{E}\left[Y|X = x\right] = \int Y \ \mathbb{P}(Y|X = x)dy. \tag{2.5}$$

This equation highlights the mean squared loss, defined as the Mean Squared Error (MSE) in Section 2.1.1, which represents the average of the squares of the errors or residuals. In this context, the mean squared loss defines the optimal conditional prediction as the conditional expected value, emphasising the importance of accuracy in this estimation. Furthermore, Eq. (2.5) is also referred to as the *regression function*, which expresses the expected relationship between the target and the predictor variables [69, p.20; 71]. Thus, the primary goal of the regression function is to estimate the unknown parameters of this relationship.

Expected Regression Function

Given a statistical learning algorithm, researchers want to estimate the regression function's expected value over every training data set:

$$\mathbb{E}\left[\hat{f}(x)|X = x\right] = \int_x \hat{f}(x) \ \mathbb{P}(X = x)dx. \tag{2.6}$$

Equation (2.6) provides the average estimate over every prediction model developed. Importantly, this calculation converges to the actual value with enough data sets based on the weak law of large numbers [72, p.335]. Unfortunately, this assumption only holds when the data is in a finite set. If X is continuous, the probability of obtaining a sample at any particular value is generally zero, which is also true about the likelihood of acquiring multiple samples at exactly the same value of x. Thus, the clear issue when estimating any function from data is that it will always be undersampled, and researchers will need to approximate in between the values in their data [69, p.23].

Researchers must also acknowledge that each y_i is a sample from the conditional distribution $Y|X = x_i$ and does not generally equal $\mathbb{E}[Y|X = x_i]$. Moreover, different methods of estimating the regression function $f(x)$ involve different interpolation, extrapolation, and smoothing processes. Each choice affects the approximation of $f(x)$ with a limited class of functions that are estimable. Furthermore, there is no guarantee that the choice will lead to a good approximation of the regression function. Although challenging, the approximation error sometimes shrinks as the size of the data set increases [69, p.23].

Expected Test Error

Given a specific prediction model $\hat{f}$, its *generalisation error* is defined as the resulting error from testing an algorithm on new data. Hence, taking a new random pair (x, y) drawn from the distribution $\mathbb{P}(X, Y)$ and determine the squared error loss:

$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2 | X = x, Y = y\right] = \int_x \int_y \left(y - \hat{f}(x)\right)^2 \mathbb{P}(X = x, Y = y) dx dy. \quad (2.7)$$

Moreover, Eq. (2.7) determines how well the model generalises on previously unseen data. This notation helps decompose the algorithm's test MSE and effectively conceptualise the bias-variance trade-off.

The First Bias-Variance Decomposition

While researchers assume that the actual regression function is $f(x)$, $\hat{f}$ develops the predictions for the data. Subsequently, the MSE at $X = x$ can be described in terms of $\hat{f}$, which is used when predictions cannot be made with f. Thus, the error $(Y - \hat{f}(x))^2$ can be expanded since the expectation of this formula is simply the MSE at x:

$$\begin{aligned}\big(Y - \hat{f}(x)\big)^2 &= \big(Y - f(x) + f(x) - \hat{f}(x)\big)^2 \\ &= \big(Y - f(x)\big)^2 + 2\big(Y - f(x)\big)\left(f(x) - \hat{f}(x)\right) + \left(f(x) - \hat{f}(x)\right)^2.\end{aligned} \tag{2.8}$$

From Eq. (1.1), $Y - f(X) = \epsilon$, since ϵ is a random variable uncorrelated with X and has an expectation of zero. The expectation in Eq. (2.8) allows us to remove the middle term since $\mathbb{E}\left[Y - f(X)\right] = \mathbb{E}\left[\epsilon\right] = 0$, while the last term remains the same since it does not contain any random quantities. The first term in the equation then becomes the variance of the irreducible error term ϵ, such that $\mathbb{V}[\epsilon] = \sigma^2(x)$:

$$\text{MSE}\left(\hat{f}(x)\right) = \sigma^2(x) + \left(f(x) - \hat{f}(x)\right)^2. \tag{2.9}$$

The prediction function does not affect $\sigma^2(x)$; instead, this term displays the difficulty of predicting Y at $X = x$. The second term in Eq. (2.9) determines the additional error that results from misrepresenting f. This breakdown is referred to as the first *bias-variance decomposition*. Here, decomposing the total MSE at x into a (squared) bias $f(x) - \hat{f}(x)$ denotes the amount by which the predictions are systematically off, and an unpredictable variance $\sigma^2(x)$, which produces statistical fluctuation around even the best prediction [69, p.24].

The Second Bias-Variance Decomposition

In Eq. (2.9), $\hat{f}$ is theoretically a single fixed function when, in reality, it is a function approximated from earlier data. Assuming the data is random, the regression function will also be random and denoted as $\widehat{M}_n$, where the subscript symbolises

the finite amount of data employed to evaluate it. From this actualisation, the $\text{MSE}\left(\widehat{M}_n(x)|\widehat{M}_n = \hat{f}\right)$ is actually being assessed, conditional on a given estimated regression algorithm. Thus, averaging over every possible training data set allows for approximating the prediction error of the method:

$$\begin{aligned}
\text{MSE}\left(\widehat{M}_n(X)\right) &= \mathbb{E}\left[\left(Y - \widehat{M}_n(X)\right)^2 | X = x\right] \\
&= \mathbb{E}\left[\left(\underbrace{Y - f(X)}_{\text{a}} + \underbrace{f(X) - \widehat{M}_n(X)}_{\text{b}}\right)^2 | X = x\right] \\
&= \mathbb{E}\left[\underbrace{\left(Y - f(X)\right)^2}_{\text{a}^2} + \underbrace{2\Big(\big(Y - f(X)\big)\big(f(X) - \widehat{M}_n(X)\big)\Big)}_{2\text{ab}}\right. \\
&\qquad \left. + \underbrace{\left(f(X) - \widehat{M}_n(X)\right)^2}_{\text{b}^2} | X = x\right]. \qquad (2.10)
\end{aligned}$$

Equation. (2.10) results in the equation $a^2 + b^2$ and adds a third term $2ab$ through factorisation. The middle term is expected to be 0 based on the assumption that the error term $Y - f(X)$ has an expected value of zero and is uncorrelated with the predictors. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\big(Y - f(X)\big)\big(f(X) - \widehat{M}_n(X)\big) | X = x\right] &= \Big(\mathbb{E}\left[Y - f(X) | X = x\right]\Big) \\
&\quad \times \left(\mathbb{E}\left[\left(f(X) - \widehat{M}_n(X)\right) | X = x\right]\right) \\
&= \Big(\mathbb{E}\left[Y | X = x\right] - f(X)\Big) \\
&\quad \times \left(f(X) - \mathbb{E}\left[\widehat{M}_n(X)\right]\right) \\
&= \big(f(X) - f(X)\big) \\
&\quad \times \left(f(X) - \mathbb{E}\left[\widehat{M}_n(X)\right]\right) \\
&= 0. \qquad (2.11)
\end{aligned}$$

This result stems from the fundamental regression assumption that the errors have an expected value of zero and are independent of the predictors, which is crucial for the model's estimations to be unbiased and consistent.

In Eq. (2.8), a^2 defines the irreducible error or the data's noise $\sigma^2(x)$. Therefore, one can focus on b^2 in Eq. (2.10) by adding and subtracting the expectation of the random regression:

$$
\begin{aligned}
&= \sigma^2(x) + \mathbb{E}\left[\left(f(x) - \widehat{M}_n(x)\right)^2 | X = x\right] \\
&= \sigma^2(x) + \mathbb{E}\big[\left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)^2 \\
&\qquad + 2\left(\left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)\left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x)\right)\right) \\
&\qquad + \left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x)\right)^2 | X = x\big].
\end{aligned}
$$

Again, the middle term can be shown to be 0:

$$
\begin{aligned}
&\mathbb{E}\left[\left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)\left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x)\right) | X = x\right] \\
&= \mathbb{E}\left[\left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right) | X = x\right] \mathbb{E}\left[\left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x)\right) | X = x\right] \\
&= \left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)\left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \mathbb{E}\left[\widehat{M}_n(x)\right]\right) \\
&= 0.
\end{aligned}
$$

Thus, the decomposition of the expected test error ends with the three remaining terms,

$$
\begin{aligned}
\mathrm{MSE}\left(\widehat{M}_n(x)\right) &= \underbrace{\sigma^2(x)}_{\text{Noise}} + \underbrace{\mathbb{E}\left[\left(f(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)^2 | X = x\right]}_{\text{Bias}^2} \\
&\qquad + \underbrace{\mathbb{E}\left[\left(\mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x)\right)^2 | X = x\right]}_{\text{Variance}} \\
&= \sigma^2(x) + \left(f(x) - \mathbb{E}_x\left[\widehat{M}_n(x)\right]\right)^2 + \mathbb{V}\left[\widehat{M}_n(x)\right]. \qquad (2.12)
\end{aligned}
$$

The first term is the variance of the entire process. This variance, or noise, represents the difference between an actual observation and the expected value of the true regression function f. The problem difficulty grows simultaneously with the difference between these two values. The second term is the bias that develops when estimating f with $\widehat{M}_n$, known as *approximation bias* or *approximation error*. Inde-

pendent of noise, the resulting Eq. (2.12) shows the bias as the difference between the prediction model's expectation and the true regression function [69, p.24].

When a prediction model consistently results in a particular solution, it is referred to as a biased estimator, and its magnitude can be determined by the difference between the actual outcome of $f(x)$ and the average result of $\widehat{M}_n(x)$. The final term in Eq. (2.12) is the variance in the statistical learning function's estimate. If one observes a random data set, the variance determines the difference between the predictions of the statistical learning algorithm and the expected regression function. Hence, even if the model is unbiased $\left(f(x) = \mathbb{E}_x\left[\widehat{M}_n(x)\right]\right)$, if there is a substantial amount of variance in the estimates, one can expect to observe significant prediction errors [69, p.24].

Approximation bias arises when a model, due to its inherent simplifications, cannot fully capture the underlying reality of the data. This is often the case with models that assume a certain form or structure of the data, which may not necessarily align with the actual, potentially more complex, structure. For instance, linear models assume a linear relationship between variables and can introduce bias if the true relationship is nonlinear. This type of bias results from the model's inability to conform to the true function that generated the data, often because of oversimplification or assumptions imposed by the model's form.

Flexible methods, while they can exhibit small approximation biases across a wide range of regression functions, often come with their own set of challenges. Primarily, reducing approximation bias typically results in an increase in estimation variance, encapsulating the essence of the *bias-variance trade-off*. Notably, in some instances, introducing a certain degree of bias intentionally can indeed reduce the overall error, as the decrease in variance may outweigh the increase in bias, a strategy known as regularization [69, p.25].

Ultimately, the approximation bias and estimation variance are heavily n-dependent. A method is defined as consistent when it recovers the actual regression function as $n \to \infty$, thus reducing its bias and variance to zero. However, consistency also depends on the method's ability to match the data-generating process, which creates another layer of the bias-variance trade-off. While the correct repre-

sentation of f is rarely found in the real-world setting, multiple consistent methods can be developed for the same problem. Furthermore, the bias and variance of a model do not have to go to zero at the same rates [69, p.25]. Thus, it is clear that the bias-variance trade-off raises serious concerns and provides distinct goals for the modelling setting where the optimal model results in low bias and low variance.

In real-world modelling settings, this trade-off is not merely a theoretical concern but a practical one that influences the reliability and predictability of models. A disregard for this balance can lead to models poorly equipped for prediction, leading to decisions that are either overcautious or reckless. Therefore, the bias-variance trade-off poses significant challenges and delineates clear objectives for modellers striving for low bias and low variance, ultimately guiding the creation of robust, reliable models instrumental in fields as diverse as healthcare, finance, and technology.

2.2 Sparse Regression

Sparse regression is an augmentation of linear regression that attempts to identify a small subset of predictors most relevant to the response variable. In many real-world scenarios, the number of potential predictors is large, and not all are expected to impact the outcome significantly. The approach employs regularisation techniques to select the most important predictors and estimate their coefficients while setting the remaining coefficients to zero. This process often produces more interpretable models, reduces overfitting, improves prediction accuracy, and automatically performs variable selection. The following section will explore some of the fundamental methods for understanding the field of sparse regression.

2.2.1 Ordinary least squares

Linear regression estimates the underlying signal in a data set, with OLS being the most common process for accomplishing this task. When developing predictions, linear regression observes n observations of an outcome variable y_i and p associated

independent variables $x_i = (x_{i1}, \ldots, x_{ip})^T$. A linear regression model assumes

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \tag{2.13}$$

where β_0 and $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ are unknown parameters and ϵ_i is an error term [52, p.2]. When assuming $\mathbf{X}$ has full column rank, the OLS approach can be solved analytically with the closed-form solution of [73, p.293; 51, p.45]

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.14}$$

OLS regression describes the fit of a linear model to a given data set by obtaining parameter estimates that minimise the RSS,

$$\hat{\beta}^{\text{OLS}} = \underset{\beta_0,\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2. \tag{2.15}$$

Equation (2.15) is a convex function because it only has one global minimum (that may be attained at more than one point) and selects $\hat{\beta}$ which minimises the objective function. Moreover, Eq. (2.15) must be carefully interpreted to understand its results. First, the intercept $\hat{\beta}_0$ is the estimated expected value of $\mathbf{y}$ when all predictors equal zero [71, p.9]. If zero is an impossible value for the predictor or predictors, there will be no physical interpretation of the intercept, meaning there can be no attempt to interpret its significance. However, when all predictors in $\mathbf{X}$ are centred to have mean zero, $\hat{\beta}_0$ becomes the sample mean of the target values. If any particular value for each predictor provides a meaningful interpretation, each variable can be centred around its specific value. When all predictors express meaningful values, the intercept becomes an estimate of the expected value of y [71, p.9].

The jth predictor's approximated coefficient $\hat{\beta}_j$ can then be interpreted to estimate the expected change in the target variable associated with one unit change in the jth predictor variable while holding all other variables in the model fixed. From this, the word "associated" implies that one cannot say that a change in the predictor "causes" a change in the target variable. Instead, one can only associate the two

variables, implying that a correlation does not indicate causation. Furthermore, the phrase "holding all other variables fixed" demonstrates the conditional relationship when fixing all other variables and assessing a particular predictor rather than the marginal relationship between a given predictor and the target while ignoring all other variables [71, p.9].

In multiple regression, explicitly including control variables in the model can help to account for their effect statistically, depending on the conditional interpretation of the coefficients. Although one cannot physically intervene in the experiment for observational data to hold other variables fixed, the multiple regression framework accomplishes this task statistically [71, p.10].

It is impossible to change one predictor and hold all others fixed from a practical viewpoint. Ideally, each coefficient can be interpreted as one that accounts for the presence of another predictor in a physical sense. However, at best, the multiple linear regression framework only approximates the underlying random process [71, p.10].

The OLS method is commonly used because of the Gauss-Markov Theorem, which states that the least squares estimates of the parameters β have the smallest variance among all linear unbiased estimates. However, a more biased estimator is known to potentially exist with a lower mean squared error based on the knowledge of the bias-variance trade-off. Therefore, any method that reduces the OLS coefficients' size or even sets them to zero may result in a biased estimate [51, p.52]. Subsequent sections explore methods that balance the complexity of the bias-variance trade-off.

Since the OLS method is convex, it always provides a solution. The Gauss-Markov Theorem explains that the OLS estimates have a low variance when $n \gg p$ and provide a good fit to the test observations. Unfortunately, because it yields a coefficient estimate for every variable in the design matrix, OLS models become more challenging to interpret as $\mathbf{X}$ obtains more predictors. The variability of the model substantially rises when n is not much larger than p, which leads it to overfit and poorly predict test data. Lastly, the results of the OLS method cannot be trusted when $p > n$ because the solution is no longer unique, forcing the variance to become ∞ [50, p.204].

2.2.2 Multicollinearity

Multicollinearity is a relevant pitfall of OLS regression, which refers to a situation where several predictor variables are strongly correlated, even if no pair of variables has an exceptionally high correlation [50, p.102]. Furthermore, structural multicollinearity occurs when the predictor variables are not independent of each other due to the nature of the data or the way the model is constructed. It can contaminate the data when the sum of the predictor variables is used as a predictor or when two or more predictor variables are derived from the same source or concept. If one suspects that the predictor variables are inherently related to each other, it is necessary to examine whether structural multicollinearity is present in the model. Ultimately, the multicollinearity problem can be challenging to detect since it is not a modelling error and typically transpires with insufficient data, meaning the number of observations is not sufficient enough to accurately estimate the model's parameters [74, p.234].

When multicollinearity occurs, the interpretation of the OLS model becomes at risk because the estimates are unstable. If the predictors covary almost perfectly, the effects of the OLS estimates can no longer be observed by increasing the value of one predictor while keeping all other variables constant. Furthermore, when highly correlated predictors exist in the regression equation, each may proxy the others without impacting the total explanatory power [74]. Therefore, this phenomenon makes it challenging to separate the individual effects of collinear variables on the response [50, p.99]. Ultimately, if two variables change together, it will likely be difficult to identify each variable's separate association with Y.

Although multicollinearity risks classical statistical inference, the regression model may still produce good forecasting results. When expanding and observing the identified system, the relationship among independent variables must hold for future data to ensure confidence in the predictions of the model. If the predictor variables remain codependent, the forecast will remain accurate as the data expands over time [74].

Multicollinearity can be measured by estimating each variable's *variance infla-*

tion factor (VIF) [74, p.248]:

$$\text{VIF}_j = \frac{1}{1 - R_j^2}, \quad j = 1, \ldots, p. \tag{2.16}$$

Here, regressing each predictor on all others in the model helps estimate the R^2 value in Eq. (2.16). VIF_j provides the proportional increase in the variance of $\hat{\beta}_j$ as opposed to what it would have been if the model contained uncorrelated independent variables [71]. At best, the minimum $\text{VIF}_j = 1$, indicating no multicollinearity for that particular predictor. However, if a given X_j has a strong relationship with other predictor variables, R_j^2 would be close to 1 and result in a high VIF_j. In regression problems, there is often a small amount of multicollinearity among predictors. Therefore, the rule of thumb that VIF_j higher than ten often indicates a problematic amount of multicollinearity for the regression model's estimation [50, p.102; 74, p.250].

Fortunately, there are a few remedies for this problem. First, removing any unnecessary variables from the model can subsequently reduce the VIF of remaining predictors. It is easiest to exclude the most suspicious variables that increase the VIF for all predictors. By removing these variables, the updated regression model typically becomes more trustworthy since the remaining VIF values are also reduced. However, this may not be a solution if Y depends on two mildly collinear predictor variables collectively but not individually [74, p.251]. For example, some settings require particular variables present to predict the true representation of Y, and using only one term may not suffice.

In settings where retaining variables is preferred rather than removing them from the regression model, a common method known as ridge regression can help reduce the effects of multicollinearity rather than having to decide which variables to exclude. Ridge regression is a traditional remedy for multicollinearity since its estimates are biased toward zero, thus reducing the regression model's variance and standard errors [74, p.279]. Section 2.2.4 further discusses ridge regression and its additional benefits for statistical modelling.

2.2.3 Linear Measurements with IID Noise

Consider a linear measurement model with added noise where

$$y_i = x_i^T \beta + z_i, \quad i = 1, \ldots, n. \tag{2.17}$$

In Eq. (2.17), z_i are assumed to be independent and identically distributed (IID), with probability density $\mathbb{P}$ on $\mathbb{R}$ [73, p.352]. Using this notation, the likelihood function is

$$\mathbb{P}_\beta(\mathbf{y}) = \prod_{i=1}^{n} \mathbb{P}(y_i - x_i^T \beta), \tag{2.18}$$

with the log-likelihood function

$$\text{loglik}(\beta) = \log\left(\mathbb{P}_\beta(\mathbf{y})\right) = \sum_{i=1}^{n} \log\left(\mathbb{P}(y_i - x_i^T \beta)\right). \tag{2.19}$$

The maximum likelihood estimate provides an optimal point for

$$\text{maximise} \sum_{i=1}^{n} \log\left(\mathbb{P}(y_i - x_i^T \beta)\right), \tag{2.20}$$

with variable β. The problem is convex if the density $\mathbb{P}$ is log-concave (log of $\mathbb{P}$ is concave [73, p.104]) and employs the ℓ_q norm with the penalty function $\log(\mathbb{P})$ [73, p.352]. Since optimisation problems aim to minimise functions, the negative log-likelihood is minimised, equivalent to maximising the log-likelihood.

In the linear regression framework, utilising the likelihood function $\mathbb{P}_\beta(\mathbf{y})$ facilitates the description of the probability that the predicted values best fit the data. The values of β often have constraints, representing prior knowledge about β or the domain of the likelihood function. These constraints can either be explicitly outlined or assigned as $\mathbb{P}_\beta(\mathbf{y}) = 0$ (for all $\mathbf{y}$) when β does not meet the prior information constraints of Eq. (2.20). For example, setting the value $-\infty$ to the log-likelihood function also sets boundaries of β that defy these previous constraints [73, p.351]. Furthermore, with prior knowledge of β, one can apply constraints to the function and redefine $\mathbb{P}_\beta(\mathbf{y})$ to be zero for particular values of β.

When estimating the value of the parameter β based on one sample $\mathbf{y}$ from the

distribution, the maximum likelihood estimation is often used and described as

$$\hat{\beta}_{\text{ml}} = \underset{\beta}{\text{argmax}} \left(\mathbb{P}_{\beta}(\mathbf{y})\right) = \underset{\beta}{\text{argmax}} \left(\text{loglik}(\beta)\right). \tag{2.21}$$

Thus, the function determines the estimates that maximise the likelihood (or log-likelihood) of predicting the observed value of $\mathbf{y}$. Many standard probability density functions are log-concave, such as the multivariate normal distribution, exponential distribution, and uniform distribution [73, p.104]. Therefore, if loglik(β) is concave for each value of $\mathbf{y}$, the constraints of β can be described as a set of linear equality and convex inequality constraints. In that case, the maximum likelihood estimation is a convex optimisation problem [73, p.352]. Therefore, convex optimisation methods can be used to compute the maximum likelihood estimate in the current setting.

In the presence of Gaussian noise with zero mean and variance σ^2, the probability density is observed as

$$\mathbb{P}(x) = (2\pi\sigma^2)^{-1/2} e^{\frac{-x^2}{2\sigma^2}}, \tag{2.22}$$

and the log-likelihood function is

$$\text{loglik}(\beta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{X}\beta - \mathbf{y}\|_2^2, \tag{2.23}$$

where $\mathbf{X}$ contains rows $x_1^T, \ldots, x_n^T$. The maximum likelihood estimate of β is, therefore, the solution of a least squares approximation, $\beta_{\text{ml}} = \text{argmin}_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2$. Since Gaussian noise frequently hinders data, the OLS model can be employed for many real-world applications. However, OLS regression does not accurately estimate when different noise types corrupt the data. For example, with Laplacian noise, the density function becomes

$$\mathbb{P}(x) = (1/2a)e^{-|x|/a}, \quad a > 0, \tag{2.24}$$

and $\hat{\beta} = \text{argmin}_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_1$ is used as the maximum likelihood estimate, which is the solution for the ℓ_1 norm approximation problem. Finally, when z_i are uniformly distributed on $[-a, a]$, the density function is

$$\mathbb{P}(x) = 1/(2a) \quad \text{on} \quad [-a, a], \tag{2.25}$$

and the maximum likelihood estimate is any β such that $\|\mathbf{X}\beta - \mathbf{y}\|_\infty \leq a$ [73, p.352]. Since Gaussian noise measurements are most commonly observed, the ℓ_2 norm can be implemented with added regularisation penalties to solve the optimisation problem.

2.2.4 Regularisation

Regularisation is a standard scalarisation method to improve the RSS of a prediction model and its interpretability. For example, adding a term to the objective function helps penalise large β, sometimes having the effect of reducing their coefficients to zero and allowing the development of a prediction model with fewer terms [50, p.203; 73, p.307; 51, 52]. The ℓ_q norms are often used to implement regularisation penalties and denoted as [75, p.29]

$$||x||_q \doteq \left(\sum_{i=1}^{n} |x_i|^q \right)^{1/q}, \quad 1 \leq q < \infty. \tag{2.26}$$

Moreover, adding the ℓ_q penalties allows us to generalise the OLS equation and view its estimates as

$$\underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}, \tag{2.27}$$

for $q \geq 1$ [51, p.72]. In Eq. (2.27), $q = 2$ results in *ridge regression*, while $q = 1$ provides the *least absolute shrinkage and selection operator* (lasso) solution. Furthermore, when $q = 0$, the regularisation penalty counts the nonzero parameters and corresponds to variable subset selection. When $q < 1$, the penalty is no longer convex, and the optimisation problem becomes NP-hard.

As λ increases in Eq. (2.27), ridge regression, the lasso, and the adaptive lasso shrink the coefficients toward zero. However, of these three methods, the lasso and the adaptive lasso automate variable selection by reducing small coefficients to exactly zero [76]. Section 2.2.5 discusses the adaptive lasso in further detail.

Many versions of Eq. (2.27) replace the standardisation factor $1/2n$ with $1/2$ or even 1, corresponding to a simple reparametrisation of λ. However, this standardisation technique makes λ values comparable for different sample sizes and beneficial

for some cross-validation techniques described in subsequent sections. As a shrinkage method, Eq. (2.27) is known to scale well to large problems because it allows one to solve the regression problem while maintaining convexity [52, p.22].

Regularisation methods fully encompass the bias-variance trade-off. The technique attempts to improve the MSE of the test set by reducing the number of variables in the prediction model, which inherently increases the bias but decreases the variance of the model. Furthermore, regularisation techniques identify a sparsity pattern to select variables with nonzero coefficients and determine the optimal $\hat{\beta}_j$. Hence, any method that reduces the size of the OLS coefficients or even sets them to zero results in a biased estimate [51].

In 1970, Arthur Hoerl and Robert Kennard [77] proposed ridge regression, an approach that employs the ℓ_2 norm to reduce the RSS by decreasing the coefficient estimates' values. The ℓ_2 norm observes the standard Euclidean length [75, p.29].

$$||x||_2 \doteq \sqrt{\sum_{i=1}^{n} x_i^2}, \tag{2.28}$$

With Eq. (2.28), the criterion in Eq.(2.27) can be written in matrix form,

$$\mathrm{RSS}(\lambda) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^T\beta. \tag{2.29}$$

To find the ridge regression solution, Eq. (2.14) can be slightly adjusted such that

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.30}$$

where $\mathbf{I}$ is a $p \times p$ identify matrix. Moreover, the ridge regression solution provides a linear function of $\mathbf{y}$ since the penalty $\beta^T\beta$ is quadratic. Before inversion, the solution makes the problem nonsingular by adding a positive constant to the diagonal of $\mathbf{X}^T\mathbf{X}$. Thus, ridge regression always provides a solution, even if $\mathbf{X}^T\mathbf{X}$ is not of full rank [51, p.64].

Ridge solutions are not equivariant under the scaling of the inputs. In Eq. (2.30), λ requires each variable to have the same magnitude (a problem that does not appear

in OLS). Therefore, the data are often standardised so that each column is centred to have mean zero and unit variance before performing penalised regression [51, p.64]. Without standardisation, the ridge regression solutions would rely on the unit measurement of a predictor [52, p.9].

As a result of standardisation, the variables are all on a comparable scale [50, p.215; 52, p.9]. Therefore, it is convenient for us to centre both the outcome values and the design matrix to remove the intercept term in the proposed optimisation method. Hence, the optimal solution $\hat{\beta}$ of the original data can be determined on the centred data. Using the standardised data, the estimates $\hat{\beta}_j$ and $\hat{\beta}_0$ are identified by

$$\begin{aligned} \hat{\beta}_j &= (s_y/s_j)\hat{\nu}_j, \quad j = 1, 2, \ldots, p, \\ \hat{\beta}_0 &= \bar{y} - \sum_{j=1}^{p} \bar{x}_j \hat{\beta}_j, \end{aligned} \tag{2.31}$$

where $\bar{y}$ and $\{\bar{x}_j\}_1^p$ are the uncentred means of the data set, and $\hat{\beta}_j$ and $\hat{\nu}_j$ are the jth estimated original and standardised regression coefficients [74, p.67; 52, p.9].

The tuning parameter λ in Eq. (2.27) directly corresponds to shrinkage because it controls the relative effect of the ℓ_q norm on the OLS equation. Therefore, when $\lambda = 0$, the problem results in the OLS equation. However, as λ increases, the coefficient estimates reduce towards zero. The optimal tuning parameter can be identified by generating different coefficient estimates for several values of λ and assessing the accuracy of each prediction model to determine the best one for estimating $\mathbf{y}$ in data [50, p.215]. Section 2.3 discusses the model accuracy metrics for machine learning methods including regularisation.

Ridge regression is a common remedy for multicollinearity since the OLS model tends to predict coefficient estimates poorly and produces a high variance when it contains several correlated variables. With multicollinearity, a substantially large positive coefficient for one variable often cancels a comparably large negative coefficient for its correlated cousin. Therefore, the added size constraint in Eq. (2.30) improves prediction results in the presence of multicollinearity by shrinking the coefficients towards each other to decrease the linear dependency in the data [51, p.63; 78, p.2]. Although the added constant value introduces some bias, ridge regression

often significantly improves the multicollinearity problem because the inverse of the algorithm's analytical solution always exists [73].

While ridge regression reduces multicollinearity, it does not perform variable selection because it still provides a coefficient estimate for all variables in $\mathbf{X}$. However, a slight modification to Eq. (2.27) allows one to improve model interpretability by forcing coefficient estimates to "shrink" to zero and promoting sparsity within $\hat{\beta}$ [50, p.204].

As Hastie, Tibshirani, and Wainwright state in *Statistical Learning with Sparsity: The Lasso and Generalizations* [52, p.xv]:

> A sparse statistical model is one having only a small number of nonzero parameters or weights.

Robert Tibshirani proposed the ℓ_1 norm as a heuristic to find a sparse solution in 1996 [73, p.309; 76]. The ℓ_1 norm attains the sum-of-absolute-values length [75, p.29].

$$||x||_1 \doteq \sum_{i=1}^{n} |x_i|, \tag{2.32}$$

When using the ℓ_1 norm as a sparse regression penalty, the resulting method becomes the popular approach known as the lasso. The lasso shares a similar penalty to ridge regression; however, the ℓ_1 norm reduces coefficient estimates to exactly zero when λ is large enough [50, p.219]. Varying λ helps identify the optimal trade-off curve between $\|\mathbf{X}\beta - \mathbf{y}\|_2^2$ and $||x||_1$, approximating the optimal trade-off curve between the OLS residual errors and the number of nonzero values in $\hat{\beta}$ [73, p.310]. Thus, the ℓ_1 norm is a proxy for the number of nonzero entries (cardinality) of $\hat{\beta}$ when applying the lasso [75, p.333].

The lasso aims to find the smallest value of λ whose solution corresponds to a subset of predictors from $\mathbf{X}$ equal to the cardinality of $\hat{\beta}$. Furthermore, as λ increases, the ℓ_1 penalty detects sparse solutions for $\hat{\beta}$ by shrinking variable coefficients with smaller residual errors to exactly zero [75, p.333; 73, p.296]. Therefore, by identifying the sparsity pattern, variable selection is performed by utilising predictors with nonzero coefficients and determining the optimal $\hat{\beta}$ that improves the interpretability of the model and minimises its RSS [73, p.310].

As a convex function, the lasso is both statistically and computationally efficient. Statistically, the ℓ_1 penalty does well to recover the underlying model when sparse. However, the ℓ_1 penalty may not be the best prediction method when the underlying signal is not sparse. Furthermore, the lasso will never select more than n parameters in its solution, resulting in a much easier computation [52, p.24]. Additionally, the lasso sometimes provides inconsistent results, especially with varying noise levels or collinear variables in the data set [79]. Here, the algorithm tends to select one variable and discard the other correlated predictors arbitrarily, forcing researchers to use alternative variable selection techniques [80].

2.2.5 The Adaptive Lasso

In 2006, Hui Zou proposed the adaptive lasso as a modification to potentially improve identification results even in the presence of multicollinearity [81]. While maintaining convexity, the adaptive lasso reduces bias in the lasso solution by adding weights w to the ℓ_1 penalty term to implement a less significant penalty on coefficients that are expected to have a larger magnitude [82]. The added weights vector allows the adaptive lasso to determine the underlying model's correct variables without prior assumptions [81, 83].

The adaptive lasso is defined as

$$\hat{\beta}^{\text{adaptive lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}, \tag{2.33}$$

where w is

$$w_j = 1/|\hat{\beta}_j(\text{OLS})|^{\nu} \tag{2.34}$$

or

$$w_j = 1/|\hat{\beta}_j(\text{ridge})|^{\nu} \tag{2.35}$$

and $\nu > 0$ is a tuning parameter [52, p.86; 51, p.92]. The weighted penalty in the adaptive lasso can be interpreted as an approximation of the ℓ_q-penalties with $q = 1 - \nu$ [52].

OLS or ridge regression is typically performed to determine w because these

techniques provide an initial coefficient vector that is not sparse, and initial estimates equal to zero make weights infinite. If $n > p$, one typically applies the OLS solution for the initial estimates of the adaptive lasso model. Since OLS estimates are not defined when $p \geq n$, ridge regression is alternatively used to identify a solution for w_j when this occurs. Furthermore, $\hat{\beta}$(ridge) provides more stable results when collinearity exists in the regression model [81].

The lasso is effective when only a few coordinates of the coefficients β are nonzero. However, like OLS, the lasso provides unstable estimates when predictors are collinear, whereas ridge regression produces more stable solutions when multicollinearity exists in the data [51]. Therefore, the adaptive lasso performs a two-stage procedure since it first performs ridge or OLS regression to identify its weights vector [84, p.44].

In the first stage of the adaptive lasso, the method applies OLS or ridge regression to obtain stable pilot estimates $\tilde{\beta}$, favouring ridge regression to reduce the effects of multicollinearity [81]. The second stage of the algorithm then applies the $\tilde{\beta}$ pilot estimates to the weights vector w and performs variable selection by solving the problem in Eq. (2.27). Here, the adaptive lasso calculates the weights vector w using pilot estimates $\tilde{\beta}$ corresponding to the optimal λ^*_{ridge} ridge regression model before identifying a separate tuning parameter $\lambda^*_{\text{adaptive lasso}}$. By fixing $\nu = 1$, the adaptive lasso develops a soft-threshold approximation to the ℓ_0-penalty. In doing so, the adaptive lasso makes Eq. (2.33) less computationally expensive since it optimises twice on a single parameter rather than simultaneously optimising over λ^*_{ridge} and $\lambda^*_{\text{adaptive lasso}}$ [84, p.44].

The adaptive lasso often yields a sparser solution than the lasso since applying individual weights to each variable places a stronger penalty on smaller coefficients, reducing more of them to zero. Here, small $\tilde{\beta}$ coefficients from the first stage of the adaptive lasso lead to a larger penalty in the second. Larger penalty terms in the second stage of the adaptive lasso result in more coefficients being set to zero than the standard lasso method. Furthermore, a smaller penalty term enables the adaptive lasso to uncover the true coefficients and reduce bias in the solution [84, 85].

The adaptive lasso is particularly useful for system identification since it obtains

the oracle property when $\tilde{\beta}_j$ converges in probability to the true value of β_j at a rate of $1/\sqrt{n}$ ($\sqrt{n}$-consistency). As n increases, the algorithm will select the true nonzero variables and estimate their coefficients equivalent to the maximum likelihood estimation [81, 85].

Instead of using the adaptive lasso, an additional threshold to the original lasso estimator can be applied as

$$\hat{\beta}_{\text{thresh},j}(\lambda, \delta) = \hat{\beta}_{\text{init},j} 1(|\hat{\beta}_{\text{init,j}}| > \delta). \tag{2.36}$$

OLS regression can then be fit to the selected variables which are given by $\hat{S}_{\text{thresh}} = \{j : |\hat{\beta}_{\text{thresh},j}|\}$ [84, p.33]. Although this thresholding and refitting method contains theoretical properties that are as good or even slightly better than the adaptive lasso, it is not commonly recommended to perform this approach over the latter. Ultimately, this thresholding technique is employed to improve model discovery for dynamical systems with the lasso and the adaptive lasso in this thesis.

2.3 Model Assessment and Selection

Model assessment and selection are two critical steps in statistical learning that involve evaluating and choosing different models to make predictions and infer relationships in the data. The model assessment process estimates a chosen model's prediction error or uncertainty on new data. Furthermore, these methods help us determine a model's performance or compare different models' ability to make predictions. Model selection refers to identifying the optimal prediction model for a given task. In doing so, different model assessment methods are used based on criteria such as accuracy, complexity, interpretability, and generalisability. The following section discusses cross-validation, Bayesian Information Criterion (BIC), and the bootstrap as standard model assessment and selection methods that allow us to expand on limited data sets and create significant distributions for prediction.

2.3.1 Cross-Validation

Cross-validation is the most common method for estimating model performance. Like the method that divides the data into a training and a test set, the *validation set approach* randomly splits the data set into three parts: a *training set*, a *validation set* or *hold-out set*, and a *test set* [50, p.176; 51, p.222]. Statistical and machine learning algorithms are initially fit to the training set before their predictions are developed for the validation set observations using the identified model. Finally, the predictions for the validation set observations enable the estimation of the test error rate of the model, typically determined using the MSE or RMSE in the regression setting.

Generally, the split depends on the size of the training sample, the signal-to-noise ratio (SNR), and the complexity of the models used to fit the data. Ideally, there would be enough data to use 50% for training and 25% each for validation and testing [51, p.222]. However, real-world applications often lack an expansive data set, so this approach typically uses a 50/50 split for only the training and validation sets.

There are two important caveats to the validation set method. Firstly, since the validation set is determined randomly, one can observe a highly variable test error rate depending on the observations in each of the two data sets. Furthermore, because the approach uses fewer observations to train and test models, the validation error rate tends to overestimate the test error rate for the fit of the model to the entire data set [50, p.200]. Therefore, one may have less confidence in the accuracy of a prediction model due to the smaller size of the data set. Moreover, the validation set approach may not be trustworthy if the distribution of test data significantly differs from the training data, potentially leading to overfitting on the validation set and poor generalisation performance on unseen data. Hence, alternative model assessment options are generally preferred to improve on these issues.

Like the validation set approach, *leave-one-out cross-validation* separates the data into two parts. However, rather than creating subsets of similar size, a single observation (x_1, y_1) can be used as the validation set while composing the training set from the remaining observations $\{(x_2, y_2), \ldots, (x_n, y_n)\}$. With leave-one-out cross-

validation, the model is fit to $n-1$ training observations and tested on the excluded observation, resulting in separate estimates of model performance. Since (x_1, y_1) was excluded from the fitting process, the MSE of the validation set is an approximate unbiased estimate of the test error. However, the MSE provides a poor estimate because it is highly variable based on a single observation. Therefore, this approach can be repeated n times to develop the average MSE of n test error estimates [50, p.200].

Leave-one-out cross-validation provides several advantages over the validation set approach. First, it is significantly less biased than the former. Since statistical and machine learning methods are fit to training sets containing $n-1$ observations, they generally do not overestimate the leave-one-out cross-validation test error rate like they often do with the validation set approach. Second, the validation set approach yields different results since it is entirely random in its training and validation split. Alternatively, leave-one-out cross-validation does not share this problem and always produces the same results because the model is fit n times [50, p.179]. Unfortunately, fitting the model n times makes the approach more computationally expensive as n increases in size.

K–fold cross-validation provides a different model assessment procedure that expands the data by creating K partitions of roughly equal-sized subsets. For example, when $K = 5$, the first fold is treated as the validation set while the model is trained to the other $K - 1$ folds of the data, allowing the prediction error of the fitted model to be observed when evaluating the kth part of the data [[51, p.241]; 50, p.203]. When performing K-fold cross-validation, the value of K determines the number of subsets created for the method. The most frequently used values include $K = 5, 10, \text{or } 20$, although the optimal value can depend on the size of the data set and the complexity of the model. The K estimates of prediction error are then combined after performing this method for $k = 1, 2, \ldots, K$, using a different fold for the validation set each time. Similarly to leave-one-out cross-validation, the optimal model is identified by calculating the minimum average MSE_K test error

estimate [50, p.181]:

$$\mathrm{CV}_{(K)} = \frac{1}{K}\sum_{i=1}^{K}\mathrm{MSE}_i, \tag{2.37}$$

As different values of K are used for the K–fold cross-validation approach, there is a clear fluctuation observed between the bias and variance in the determined model.

In addition to reducing bias and variance, K–fold cross-validation is often preferred over the previously discussed approaches because it provides a computational advantage. With K–fold cross-validation, a set of candidate models with varying flexibility is reviewed, and the model corresponding to the minimum test error is identified as optimal. When performing cross-validation, one aims to determine how well they can expect a given statistical learning model to perform on independent data, i.e., the actual estimate of the test MSE of interest [50, p.183].

2.3.2 Bayesian Information Criterion

BIC is an alternative method for model assessment that applies the Bayesian argument to a list of candidate models and determines the one with the highest posterior probability [51, p.234; 70, p.147]. This method assesses the maximum likelihood estimation to select the model that best fits the data,

$$\mathrm{BIC} = -2 \cdot \mathrm{loglik} + \log(n) \cdot p, \tag{2.38}$$

where p is the number of selected predictors. When assuming that the model contains Gaussian errors, one expects prior knowledge of the variance, σ^2, then $-2\cdot$loglik is equivalent to $\sum_i (y_i - \hat{y}_i)^2/\sigma^2$ for the squared error loss [51, p.233; 50, p.234]. The BIC equation can then be transformed to

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + \log(n) \cdot p\hat{\sigma}^2\right). \tag{2.39}$$

The result from Eqs. (2.38) and (2.39) is often smaller for models with low test errors. Hence, the optimal model corresponds to the one with the minimum BIC value [50, p.234].

The BIC approach promotes sparsity because it places a heavier penalty on

models with more variables and avoids complex models as n increases [70, p.147]. Equations. (2.38) and (2.39) can be generalised for variable selection techniques as

$$\mathrm{BIC}(\hat{f}) = \mathrm{MSE} + \frac{\log(n)}{n} \cdot \widehat{df}(\hat{f}). \tag{2.40}$$

where $\widehat{df}(\hat{f})$ is the number of selected variables found by a given variable selection algorithm [86–88]. The optimal λ value can be determined in Eqs. (2.27) and (2.33) by

$$\lambda(\mathrm{optimal}) = \underset{\lambda}{\mathrm{argmin}}\, \mathrm{MSE} + \frac{\log(n)}{n} \cdot \widehat{df}(\lambda). \tag{2.41}$$

where $\widehat{df}(\lambda)$ is the number of selected variables in the model corresponding to λ. The λ(optimal) is identified by calculating the BIC for a set of models with unique values of λ. Furthermore, one selects the model whose λ corresponds to the minimum BIC, increasing the probability that our model most accurately predicts $\mathbf{y}$.

BIC is asymptotically consistent and is much less computationally expensive than cross-validation as $n \to \infty$ [51, p.235; 87]. Additionally, if the true model exists among the evaluated prediction models, the probability that the model with the minimum BIC is the correct model converges to one as n increases. However, BIC struggles when n is finite and p is large because it reduces bias and often includes irrelevant variables for prediction [70, p.147].

2.3.3 Bootstrap Sampling

The bootstrap is a standard resampling method that assesses various prediction models [51, p.249; 50, p.209]. As a method that applies to multivariate systems, the bootstrap randomly draws samples with equal probability and replacement to approximate the entire population and create an empirical distribution function [89, 90]. The bootstrap accomplishes this task by randomly creating new samples with replacement when only a finite population exists in the original data set. The term "with replacement" implies that the same observation can be selected more than once in the bootstrap data set. Furthermore, each bootstrap sample contains the same number of observations as the original data set. Therefore, generating B samples renders many data sets fully representing the observations.

The statistics of each bootstrap sample can then be evaluated, and the prediction model can be examined over many data sets. For example, the bootstrap can be used to estimate the standard error and bias of the developed model. Here, the original data set acts as the test set, and the B bootstrap data sets act as the training samples. However, when $B < 50$, the bootstrap standard error and bias estimates are not always reliable because there is an overlap in observations between training and test data sets that sometimes leads to overfitting predictions. As B increases, the standard error and bias of the estimates are subsequently reduced. Ultimately, $B > 200$ is rarely necessary to determine accurate standard error measures for a prediction model [91, p.215; 89, p.52; 51, p.250].

Alternatively, bootstrap confidence intervals can be calculated for variable selection. However, many more bootstrap samples ($B \geq 2000$) are needed to develop robust confidence intervals for model selection [91, p.205; 89, p.52]. Therefore, with B, one develops quantiles from the empirical distribution of bootstrap coefficient estimates that denote the upper and lower bounds of the confidence intervals. The desired $100(1-\alpha)\%$ accuracy measure is then used to calculate the confidence interval regions, where α denotes each variable's significance level [90, p.4]. The lower bound estimate, $CI_{\text{lo}} = [B\alpha/2]$, is the integer part of $B\alpha/2$, while the upper bound estimate is $CI_{\text{up}} = B - CI_{\text{lo}} + 1$ [90, p.24]. Bootstrap confidence intervals provide uncertainty measures to identify a model consisting of variables whose confidence intervals do not cross zero and whose point estimates fall within their confidence intervals [45, p.510; 50, p.82; 92].

2.4 Bayesian Regression

Bayesian regression provides an alternative to frequentist methods like cross-validation, bootstrap sampling, and OLS regression, which addresses model uncertainty in the regression problem by treating the parameters as random variables with prior distributions, which are updated to obtain the posterior distributions of these parameters given the data [93]. In this setting, y is assumed to be drawn from a probability distribution rather than estimated as a single value. Thus, the model

for Bayesian regression assumes the sampled response is from a normal distribution, characterized by mean and variance:

$$y \sim N(\beta^T X, \sigma^2 I). \tag{2.42}$$

Equation 2.42 describes the mean for linear regression as the transpose of the coefficient matrix multiplied by the design matrix [93]. Moreover, the variance is defined as the square of the standard deviation σ multiplied by the identity matrix for a multi-dimensional formulation of the prediction model. The choice of prior distributions for the model parameters, β and σ^2, is an important aspect to consider in Bayesian regression, as they affect the posterior distribution, reflecting prior knowledge about the parameters, or non-informative, representing a lack of prior information.

Given the model, the method determines the posterior distribution for its parameters [93]. The coefficients or weights originate from posterior model probabilities, which results from Bayes' theorem [94]:

$$\mathbb{P}\left(\beta \mid y, \mathbf{X}\right) = \frac{\mathbb{P}\left(y \mid \beta, \mathbf{X}\right) \mathbb{P}\left(\beta\right)}{\mathbb{P}(y \mid \mathbf{X})}. \tag{2.43}$$

Here, $\mathbb{P}(\beta \mid y, \mathbf{X})$ denotes the probability distribution of the model parameters given the dependent variable and terms in the design matrix [95]. Furthermore, this posterior distribution is obtained by updating the prior distribution with the observed data, reflected in the likelihood of the data, $\mathbb{P}(y \mid \beta, \mathbf{X})$ [93]. With this approach, assuming the priors are non-informative or weakly informative, as the number of observations increases, the likelihood function becomes more powerful than the prior probability, and the coefficients converge to the OLS estimates.

One challenge in Bayesian regression is computing the posterior distribution, especially when the number of parameters or data points is large. However, Markov chain Monte Carlo (MCMC) methods provide a solution for this problem by generating samples from the posterior distribution, which can then be used to approximate various quantities of interest. A popular MCMC method is the Gibbs sampler, which iteratively samples from the full conditional distributions of each parameter given

the others [96, p.42]. Another widely used method is the Metropolis-Hastings algorithm, which generates a Markov chain whose stationary distribution converges to the target posterior distribution [96, p.44]. MCMC methods have been extensively used in Bayesian regression and other Bayesian modelling applications, allowing practitioners to overcome computational challenges and make inferences based on the posterior distribution of model parameters [96, p.37].

Importantly, Bayesian inference facilitates the development of posterior uncertainty intervals, also known as credible intervals [97]. Credible intervals can follow two forms: equal-tailed, where the probability mass is evenly distributed on both sides of the interval, and highest posterior density intervals, which include the most probable values of the parameter [97]. These intervals provide the likelihood that a parameter estimate encompasses the true parameter value based on the data and its prior distribution. Unlike the frequentist confidence intervals discussed in Section 2.3.3, the posterior interval provides a valid statement that given the data and model, the probability $\mathbb{P}$ indicates the likelihood of a parameter value being within its $100\mathbb{P}\%$ bound [97]. This feature of Bayesian regression helps provide a more comprehensive understanding of model uncertainty and enhances the overall inference quality.

2.5 Clustering Methods

Clustering enables researchers to partition the data into distinct subgroups or clusters so that the observations within each group are similar. Meanwhile, these methods guarantee different observations between groups [50, p.516]. While supervised learning problems aim to predict a particular outcome, unsupervised learning helps discover structure within our data.

2.5.1 K-means Clustering

K-means clustering is a straightforward and refined process for partitioning a data set into K distinct, non-overlapping clusters. The K-means clustering method is performed by specifying the desired number of clusters K, then allocating each

observation to precisely one of the clusters with the K-means algorithm. By applying a simple and intuitive mathematical problem, $C_1, \ldots, C_K$ denotes sets containing the indices of the observations in each cluster. Here, at least one of the observations belongs to at least one of the K clusters, and the clusters are non-overlapping. Therefore, no observation belongs to more than one cluster [50, p.517].

K-means clustering seeks to reduce the within-cluster variance as much as possible, where the within-cluster variance for cluster C_K is a measure $W(C_K)$ of the amount by which the observations within a cluster differ. Thus, the approach aims to solve the problem

$$\underset{C_1,\ldots,C_K}{\text{minimise}} \left\{ \sum_{k=1}^{K} W\left(C_k\right) \right\}. \tag{2.44}$$

Here, Eq. (2.44) enables the method to partition the observations into K clusters such that the total within-cluster variance over all clusters is minimal. To solve Eq. (2.44), K-means clustering often uses the squared Euclidean distance

$$W\left(C_k\right) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2, \tag{2.45}$$

where $|C_k|$ denotes the number of observations in the kth cluster. Thus, the within-cluster variance for the kth cluster is the sum of all pairwise squared Euclidean distances between each observation in the kth cluster, divided by the total number of observations in the kth cluster [50, p.518]. Eqs. (2.44) and (2.45) are then combined to derive the K-means clustering optimisation problem [50, p.518]:

$$\underset{C_1,\ldots,C_K}{\text{minimise}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 \right\}. \tag{2.46}$$

To solve Eq.(2.46), a number from 1 to K is randomly assigned to each of the observations, serving as initial cluster assignments. Next, the cluster centroid is determined for each K cluster, where the centroids are computed as the mean of the observations assigned to each cluster. Here, the kth cluster centroid is the vector containing the p feature means for the observations in the kth cluster. Then, the Euclidean distance is used to assign each observation to the cluster whose centroid

is closest and repeat this process until the cluster assignments stop changing [50, p.519].

In practice, the within-cluster dissimilarity $W(C_K)$ is often used to determine the optimal K^* for the algorithm, which generally decreases with increasing K. Here, the solution criterion will tend to decrease substantially with each successive increase in the number of specified clusters, $W(C_{K+1}) \ll W(C_K)$, as the natural groups are successively assigned to separate clusters. Typically, there will be a sharp decrease in the successive differences in criterion value, $W(C_K) - W(C_{K+1})$, at $K = K^*$ [51, p.518]. A heuristic approach can then be applied to obtain an estimate of $\hat{K}^*$ as the optimal number of clusters, which can often be used to identify through an elbow in the plot of $W(C_K)$ as a function of K.

2.5.2 Hierarchical Clustering

Although K-means clustering has many practical applications, it has the disadvantage of pre-specifying the number of clusters K. Hierarchical clustering provides an alternative approach that does not require a commitment to any choice parameters. Furthermore, the method produces a tree-based representation of the observations known as a dendrogram. Agglomerative clustering is the most common hierarchical clustering method, referring to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk [50, p.521]. When applied, each leaf of the dendrogram represents one of the observations in the data set. As one moves up the tree, some leaves start to fuse into branches corresponding to similar observations. Furthermore, as one continues up the tree, the branches eventually fuse with leaves or other branches.

In hierarchical clustering, the earlier these fusions occur, the more similar groups or observations are with one another. Alternatively, observations that fuse later can be quite different, and the height of the dendrogram allows us to determine how different two observations are from one another. To determine the number of observations based on the dendrogram, one can make a horizontal cut across the dendrogram and interpret the distinct sets of observations beneath the cut as clusters. Therefore, the height of the cut acts like K in K-means clustering and

controls the number of clusters obtained [50, p.524].

To perform hierarchical clustering, the dissimilarity measure between each pair of observations is often defined using Euclidean distance. The algorithm then starts from the bottom of the dendrogram and treats each n observation as its cluster. First, the two most similar clusters are fused to contain $n-1$ clusters, and then the following two clusters most similar to each other are fused again, providing $n-2$ clusters. The algorithm repeats this process until all observations belong to one cluster and the dendrogram is complete [50, p.525].

Although this algorithm provides a simple clustering approach, the concept of dissimilarity between observations needs to be extended to a pair of groups of observations. This problem can be achieved through linkage, which defines the dissimilarity between two groups of observations, commonly developed as complete, average, single, and centroid [50, p.525]. The choice of dissimilarity affects the resulting dendrogram significantly and requires understanding the data type and scientific question. Ultimately, this measure varies for different problems, and hierarchical clustering becomes more challenging as the size of our data increases.

2.5.3 Otsu's Method

Thresholding is a prolific method in image processing that can be viewed as a statistical decision theory problem aiming to minimise the average error caused by assigning pixels to two or more groups or classes [98, p.747]. Although this is a common approach to image thresholding, it develops assumptions that are often complex and only sometimes well-suited for real-world applications [98, p.747]. Otsu's method provides an alternative thresholding method by developing an optimum approach that maximises the between-class variance, another well-known measure used in statistical discriminant analysis [98, p.748].

The theory behind this process is that well-separated classes should have distinct measurement values and that the optimal threshold gives the best separation between classes regarding their magnitudes. With Otsu's method, properly thresholded classes should be distinct concerning the intensity values of their pixels [98, p.748]. Conversely, a threshold producing the best separation between classes in

terms of their magnitudes will best represent the data. Additionally, Otsu's method is entirely derived from computations performed on an image's histogram, a readily available one-dimensional array [98, p.748].

Otsu's method is commonly used for converting a grayscale image to monochrome [99]. This technique involves iterating through all the possible threshold values and calculating a variance for the pixel intensities on each side of the threshold, i.e., the pixels in the foreground and background. In this context, 'variance' specifically refers to the statistical measure of the spread among the pixel intensities within each class, foreground and background, and the method seeks to minimise this within-class variance. By maximising the between-class variance, Otsu's method effectively minimises the within-class variance of thresholded black and white pixels. First, the technique proposes a criterion for maximising the modified between-class variance equivalent to the usual between-class variance for image segmentation [99]. Then, with the new criterion, the method applies a recursive algorithm to find the optimal threshold efficiently.

The classic thresholding technique performs cluster-based image thresholding by diminishing the grayscale image to a binary or threshold image [99]. The algorithm considers that the image contains two classes of pixels backing a bimodal histogram. Furthermore, the method then enumerates the optimum threshold disconnecting the two classes, minimising their combined within-class variance. Thus, determining a minimal within-class variance makes the classes more distinct and easier to separate [99, 100].

Otsu's thresholding method performs automatic binarisation-level decisions based on the shape of the histogram developed with the one-dimensional array [99]. The algorithm assumes that the image comprises a foreground and background class. Otsu's method computes the optimal threshold value that minimises the weighted within-class variances of these two classes, distinguishing a given image. Furthermore, the method relies on the mathematical proof that minimising the within-class variance is the same as maximising the between-class variance [99].

In Chapter 4, Otsu's method will be instrumental in determining an initial threshold value between OLS coefficients for the sparse identification of nonlinear

dynamics algorithm (discussed in Section 2.9). The innovative application of Otsu's method in this context underscores its versatility and its unexplored potential in areas beyond its conventional usage.

2.6 System Identification

Dynamical systems use mathematical models to describe the temporal evolution of a physical process and exist throughout biology, engineering, and mathematics. Since continuous-time dynamic models necessitate derivatives to illustrate their expansion rate, these systems are often expressed with sets of ordinary (ODEs) and partial differential equations (PDEs). These equations enable researchers to analyse climate and stock market trends, fluid flow dynamics, and epidemiological models. Dynamical systems are described as [18, p.230; 46; 72, p.64]

$$\frac{d}{dt}x_j(t) = \dot{x}_j(t) = f_j(x(t)), \qquad j = 1, \ldots, m. \tag{2.47}$$

At time t, the vector $x(t) = (x_1(t) \;\; x_2(t) \;\cdots\; x_m(t))^T \in \mathbb{R}^m$ depicts the state space of the system, where m is the state space dimension, while the function $f(x(t)) : \mathbb{R}^m \to \mathbb{R}^m$ provides the restrictions that define the evolution of the system in time [101]. Equation (2.47) examines nonlinear ODEs both qualitatively and geometrically. Instead of determining exact or approximate functions for particular solutions, this model proves the occurrence, stability, and global behaviour of various solutions [102, p.383].

Moreover, the jth equation of f can be approximated symbolically by

$$\begin{aligned} \dot{x} = f(x) &\approx \sum_{i=1}^{p} \theta_{\mathrm{F}}(x)_i \beta_i \\ &= \theta_{\mathrm{F}}^T(x)\beta, \end{aligned} \tag{2.48}$$

where $\beta = (\beta_1, \ldots, \beta_p)^T$ is a sparse coefficient vector with elements that represent the parameters of the system. In this setting, $\theta_{\mathrm{F}}(x)$ is a feature vector containing p symbolic functions, each representing an ansatz that can be used to describe the

dynamics.

To develop a basis for data-driven, $x(t)$ is expanded from measurements taken at times $t_1, t_2, ..., t_n$, creating a state matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times m}$ [46]. Throughout this work, the Savitzky-Golay filter [103], described Section 2.8, is employed to smooth each column $\mathbf{x}_j = SG(\tilde{\mathbf{x}}_j)$ and calculate the derivative $\dot{\mathbf{x}}_j$. The smoothed state matrix $\mathbf{X}$ is then consolidated to build the block design matrix $\mathbf{\Theta}(\mathbf{X}) \in \mathbb{R}^{n \times p}$:

$$\mathbf{\Theta}(\mathbf{X}) = \begin{pmatrix} | & | & | & & | & | \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{[2]} & \cdots & \mathbf{X}^{[d]} & \mathbf{\Phi}(\mathbf{X}) \\ | & | & | & & | & | \end{pmatrix}, \tag{2.49}$$

where $\mathbf{X}^{[i]}$ for $i = 1, \ldots, d$ is a matrix whose column vectors denote all possible monomials of order i in $x(t)$ [46], and

$$\mathbf{\Phi}(\mathbf{X}) = \begin{pmatrix} | & | & | & | \\ \phi_1(\mathbf{X}) & \cdots & \phi_g(\mathbf{X}) & \cdots \\ | & | & | & | \end{pmatrix} \tag{2.50}$$

for $i = 1, \ldots, g$ can contain a set of nonlinear basis functions such as trigonometric, logarithmic, or exponential [46].

Each $\dot{\mathbf{x}}_j$ are then combined to develop the derivative matrix $\dot{\mathbf{X}}$ and a linear regression is formulated to solve the system identification problem:

$$\dot{\mathbf{X}} = \mathbf{\Theta}(\mathbf{X})\mathbf{B} + \mathbf{E}, \tag{2.51}$$

where $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $\mathbf{E} \in \mathbb{R}^{n \times m}$ are the coefficient and residual error matrices, respectively. Methods can then be applied to extract the jth column of $\dot{\mathbf{X}}$ and $\mathbf{B}$ from Eq. (2.51) to determine the active terms for the jth equation in Eq. (2.47). Thus, for the remaining chapters of this thesis, $\mathbf{X}$ will represent the state space matrix and $\mathbf{\Theta}(\mathbf{X})$ will define the block design matrix containing the candidate library for system identification.

In most settings, the rectangular design matrix of candidate functions, Eq. (2.49), is often defined as either *overdetermined* or *underdetermined*. Here, an overdeter-

mined system contains more equations than unknowns ($n > p$), while an underdetermined one has fewer equations than unknowns ($p > n$) [102, p.273]. When observing overdetermined systems, researchers aim to minimise the error towards zero. However, since the error is typically never zero, researchers commonly apply the least squares problem to each column of Eq. (2.47) because the approach is numerically solvable and provides an estimate for the system.

When $n > p$ and $\boldsymbol{\Theta}(\mathbf{X})$ has full rank, there are infinitely many solutions to the system, and one typically chooses the solution with the minimal ℓ_2 norm. Conversely, researchers can employ the ℓ_1 norm, or sparse regression, to develop a solution for underdetermined systems, which is important in compressed sensing problems [102, p.274]. Uncovering the underlying equations of dynamical systems facilitates forecasting, predicting, controlling, and analysing their development and structural stability. Determining the equations and variables that make up dynamical systems is a long-sought-after goal in the previously mentioned fields and is better known as system identification.

System identification generates models to explore dynamical systems in real-world data. A crucial aspect often underscoring the robustness of such explorations is the role of boundary conditions in shaping the system's behaviour and responses. Boundary conditions delineate the limits within which systems evolve and interact, fundamentally influencing the mathematical frameworks employed in system identification. They aid in formulating precise ODEs and PDEs that capture the intricate dynamics of systems, whether in climate trends, fluid dynamics, or stock market behaviours.

When these models expand over time, researchers can detect significant changes in the system, observe adjustments based on parameters that have been identified, and implement control methods for stabilisation [102, p.540]. System identification models have two components. The first component, the *deterministic* model, provides the mathematical description of the cause-effect (input-output) relationships in data. Here, boundary conditions play a pivotal role in demarcating the operational scope within which the input variables, or independent variables, affect the output variables or dependent variables. These conditions are particularly critical when

the input sets contain variables that can be controlled and manipulated, known as probe signals, and those that are unalterable but measurable, known as measured disturbances. Since the profiles of the inputs of deterministic models are often known, these models can be used to explain processes whose physics are accurately known [72, p.62]. System identification aims to identify a model that describes the input-output relationship because it explains the dynamic system in the data set.

The second component of system identification models is a statistical-plus-mathematical description of the uncertainties in the data set, the *stochastic* model. In this realm, boundary conditions help quantify and manage the unpredictability, offering a structured approach to handling external disturbances and unmeasured dynamics contributing to system uncertainties. The efficacy of the stochastic model in depicting observation and modelling errors and process uncertainties is considerably amplified when these boundary conditions are meticulously accounted for and integrated [72, p.3].

While these two components of system identification allow for the description of dynamical systems, the accuracy and reliability of the deterministic model heavily depend on the assumptions the stochastic model represents. Unlike the optimal deterministic model that is developed in a *functional* sense, the stochastic model faces the challenge that the observed response is one of several possible target variables since the optimal model is fit using a *statistical* framework. Thus, this model is predominantly referred to as a *time-series model*. Stochastic modelling theory is commonly employed to forecast changes in any process where one cannot associate any external cause or the cause is not measured or known. While inputs to deterministic models are known, inputs to stochastic models are random signals that assume values from a probability distribution. Stochastic models are determined from data since they contain fixed statistical properties [72, p.63].

Ultimately, since an element of uncertainty always exists, no system is truly deterministic. However, from an engineering perspective, systems are deterministic if the degree of predictability is very high. Therefore, composite models are usually built in identification, i.e., a deterministic plus stochastic model [72, p.63]. Furthermore, identifying a parsimonious model helps distinguish a system's behaviour

within data. Statistical and machine learning methods commonly determine the critical inputs for system identification models.

Nonlinear system identification continues to emerge as an area of growing interest [18, 46, 47, 60, 67, 104–107]. The nonlinear nature of these systems makes their identification difficult as they expand. Nonlinear system identification methods often use regression to observe a dynamical system [72, p.777]. In addition, engineers often use black-box models, such as deep learning, to represent dynamical systems. These models specify the functional relationships between input and output measurements [108]. Deep learning appeals to many engineers due to its ability to represent a complex system from high-dimensional data [60, 109].

Within deep learning, neural networks improve the ability to perform system identification. Moreover, recurrent neural networks enable engineers to forecast and reconstruct complex systems [60, 109–111]. Several layers of prediction often compose neural network models, developing feedback paths from their output layers using previous predictions for their input layers [72, p.778]. Black-box models are customary because they provide user results without focusing on the model's mathematical structure. However, black-box models face serious drawbacks because they become more challenging to interpret as the system becomes increasingly complex.

When the model structure is unknown, methods that observe an overcomplete basis of state variables, approximate model dynamics, and remove terms that do not influence the dynamics are becoming more prevalent [64]. Engineers frequently employ symbolic regression [53, 56, 112], polynomial nonlinear autoregressive moving average models with exogenous inputs methods [113–115], and sparse regression [46, 62, 64] to perform system identification and represent the governing equations in their data.

Recent methods deploy sparse regression techniques due to their ability to accurately identify the underlying equations of the nonlinear ODE and PDE systems [46, 67, 105–107, 116–119]. It has also been extended to group sparsity problems to develop parsimonious representations of dynamics for ODEs [120] and parametric PDEs [104]. Sparse regression allows scientists to identify equations and distinguish their underlying parameters from data, providing a unique approach to

system identification. Moreover, this approach extends the field of automation by allowing researchers to determine transparent models efficiently, containing the most significant variables for prediction.

Of course, selecting a favourable model depends on the data measurements that reflect the system's behaviour. System identification methods typically assume stationarity. Thus, data pre-processing is essential to ensure that drifts, trends, and other non-stationarity properties do not corrupt the data set [72, p.21]. These *noisy* data measurements compromise the results of the model identification process and lead algorithms to select variables that do not accurately represent the system. In the presence of contaminated data, denoising methods can improve modelling for high-dimensional ODEs by smoothing the signal before performing identification [103, 120, 121]. While most of these methods are theoretical, some techniques provide a distinct representation of the numerical derivative and enhance the system identification approach.

2.7 Signal-to-Noise Ratio

To observe the impact noise magnification has on system identification, researchers quantify the quality of the noise-contaminated signal. If $\mathbf{x}_s(n)$ are real-valued signal samples and $\mathbf{x}_n(n)$ are real-valued noise samples, researchers measure or estimate the signal-power-to-noise-power ratio (SNR) of a signal $\mathbf{x}(n) = \mathbf{x}_s(n) + \mathbf{x}_n(n)$ as [122, p.D-13]

$$\text{SNR} = \frac{\text{Signal power}}{\text{Noise power}} = \frac{\frac{1}{n}\sum_{n=0}^{n-1}\left[x_s(n)\right]^2}{\frac{1}{n}\sum_{n=0}^{n-1}\left[x_n(n)\right]^2}. \tag{2.52}$$

When the variance of $\mathbf{x}_s(n)$ and $\mathbf{x}_n(n)$ is known, the SNR of the fluctuating (AC) portion of a signal is defined as

$$\text{SNR} = \frac{\text{Signal variance}}{\text{Noise variance}} = \frac{\sigma_s^2}{\sigma_n^2}. \tag{2.53}$$

The SNR is commonly expressed in decibels (dB):

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} (\mathrm{SNR}) \, \mathrm{dB}. \tag{2.54}$$

Furthermore, if the RMSE values of $\mathbf{x}_s(n)$ and $\mathbf{x}_n(n)$ are known, one can provide the SNR of the signal as

$$\mathrm{SNR}_{\mathrm{dB}} = 20 \log_{10} \left(\frac{\text{Signal RMSE}}{\text{Noise RMSE}} \right) \mathrm{dB}. \tag{2.55}$$

Since the ratio in Eq. (2.55) is expressed in terms of amplitude rather than power, the factor of 20 is used to compute $\mathrm{SNR}_{\mathrm{dB}}$ based on RMSE values [122, p.D-12]. For many data sets, the standard deviation of a signal is often known and therefore an SNR is developed in the data by corrupting the state space matrix $\mathbf{X}$ with zero-mean Gaussian noise $\mathbf{Z} \sim \mathcal{N}(0, \sigma_{\mathbf{Z}}^2)$, such that

$$\mathrm{SNR} = 20 \log_{10} \left(\frac{\sigma_{\mathbf{x}_j}}{\sigma_{\mathbf{z}_j}} \right), \qquad j = 1, \ldots, m. \tag{2.56}$$

The standard deviation $\sigma_{\mathbf{z}_j}$ of each column of $\mathbf{Z}$ is determined by

$$\sigma_{\mathbf{z}_j} = \sigma_{\mathbf{x}_j} \cdot 10^{-\frac{\mathrm{SNR}}{20}}, \qquad j = 1, \ldots, m, \tag{2.57}$$

and can be implemented as

$$\tilde{\mathbf{x}}_j = \mathbf{x}_j + \mathbf{z}_j, \qquad \mathbf{z}_j \sim \mathcal{N}(0, \sigma_{\mathbf{z}_j}^2), \qquad j = 1, \ldots, m, \tag{2.58}$$

where $\tilde{\mathbf{x}}_j$ represents each column of the system corrupted by noise $\tilde{\mathbf{X}}$ [122, 123]. The following chapters employ this approach to observe the capability of each method in performing system identification on noise-contaminated data sets.

2.8 Savitzky-Golay Filter

When performing system identification, one must be careful in determining the correct method for approximating the derivative of a signal. With prior knowledge

of the underlying function, the equations can be used to measure the derivative and develop the problem in Eq. (2.51). However, researchers typically do not have access to the governing equations and often approximate the derivative numerically from the state vector $x(t)$. Throughout this thesis, the Savitzky-Golay filter is employed to approximate the numerical derivative, a standard tool when assuming noise exists in the data set [103, 124–126].

Data smoothing is based on the idea that the measured variable changes slowly and is affected by random noise [124, p.766]. In this setting, researchers can replace each data point with the local average of surrounding data points. Furthermore, averaging nearby points can reduce noise without introducing significant bias to the obtained value since these nearby points measure almost the same underlying value [124, p.767]. The low-pass Savitzky-Golay filter is well-adapted for data smoothing and does not rely on a definition in the Fourier domain. Instead, the method is derived from a specific formulation of the data smoothing problem in the time domain [124, p.767]. This filtering approach was initially applied to noisy spectrometric data to display the relative width and height of spectral lines [124, p.767].

Savitzky-Golay filtering develops a generalised moving average technique to smooth a noisy signal before differentiation. The method initially determines coefficients by performing an unweighted linear regression with a polynomial to smooth the data without distorting the signal tendency and maintaining the signal's shape [103, 125]. Furthermore, the Savitzky-Golay filter performs this least squares polynomial regression process on successive subsets, or windows, of adjacent data points and develops a smoothed generalisation of the noisy data set [103, 125]. However, if the data's underlying function is constant or changes linearly with time, no bias is introduced into the result. Alternatively, if the underlying function has a nonzero second derivative, a bias is introduced to the signal.

Typically, researchers aim to find a sparse solution to an overdetermined system with noise, which is why the Savitzky-Golay filter provides a convenient smoothing approach to perform system identification. Many scientists and engineers use the Savitzky-Golay filter because the local regression process resembles the waveform

of an oversampled signal corrupted by noise. Moreover, the filtered signal often maintains the width and height of the peaks in the signal waveform [125]. With the Savitzky-Golay filter, two parameters: window length and polynomial order are employed to smooth the data [103, 125]. When using these parameters, the window length must be odd to centre around a given point, and the polynomial order must be less than the window length.

The smoothed output calculation acquired by sampling the fitted polynomial equals a fixed linear combination of the local set of input samples at each position. The filter's derivative approximates $\dot{x}(t)$ after fitting the regression model to the data [103]. Therefore, researchers typically assume they are investigating noise-contaminated data and apply the Savitzky-Golay filter to the state variables to provide a smoothed signal before differentiation.

2.9 Sparse Identification of Nonlinear Dynamics

The recently proposed framework known as the *sparse identification of nonlinear dynamics* (SINDy) provides an alternative machine learning-based method for developing parsimonious models to describe the underlying dynamics of a system from observational data [46, 47]. SINDy accomplishes this task by adopting a sequential thresholded least squares algorithm that relaxes the ℓ_0 norm as a shrinkage penalty to the OLS equation [46, 67, 105–107, 116–119]. Thus, to identify the dynamical system in Eq. (2.51), SINDy adds the ℓ_q-penalties to the OLS estimate:

$$\underset{\mathbf{B}}{\operatorname{argmin}} \left\| \dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\mathbf{B} \right\|_2^2 + \lambda |\mathbf{B}|_0. \tag{2.59}$$

The ℓ_0 norm in Eq. (2.59) promotes sparsity in the OLS equation since it quantifies the number of nonzero elements in the coefficient vector, encouraging solutions that have a small number of nonzero elements [75].

The sequential thresholded least squares algorithm employed by SINDy allows the algorithm to solve the problem in Eq. (2.59) by implementing a hard-thresholding penalty, as opposed to the soft-thresholding algorithms described in Eq. (2.27), requiring only a single parameter λ to determine the degree of sparsity [46]. This

sparsity constraint ensures that the identified model is parsimonious and physically interpretable, which helps to reduce the risk of overfitting the data. The process is repeated recursively on the remaining nonzero coefficients until it arrives at a prediction model containing only estimates greater than that threshold value, which has proven computationally efficient, converging rapidly to a sparse solution while requiring only a small number of iterations [46]. Furthermore, when applying sequential thresholded least squares, SINDy determines a final prediction model with OLS regression, meaning the estimates of the identified variables are unbiased. However, sequential thresholded least squares may require further investigation to identify a sparse basis for its prediction model when highly correlated predictors exist in the design matrix since OLS coefficients are unstable when this occurs.

To combat the issue of multicollinearity, the SINDy framework has also developed a sequential thresholded ridge regression algorithm. This modified algorithm substitutes OLS with ridge regression to calculate its final estimates while implementing regularisation [47]. Here, the added regularisation in the thresholding step eliminates unnecessary terms and noisy features, which improves the stability of the model by reducing overfitting. However, since ridge regression shrinks coefficients toward zero, the sequentially-thresholded ridge regression method imposes an inherently biased process for identifying underlying systems.

The SINDy algorithm can be used to identify high-dimensional dynamical systems, such as fluid flow dynamics past a cylinder, nonlinear optics, and plasma physics [18, p.250]. Several authors have developed a constrained SINDy optimisation problem by consolidating physical constraints and symmetries in the equations. These constraints can promote stability, which improves energy preservation on the quadratic nonlinearities in the Navier-Stokes equations imposed for fluid system identification [127]. When examining fluid flows, SINDy can identify high-dimensional dynamical systems models from a few physical sensor measurements, such as lift and drag measurements on a cylinder. There have also been generalisations of SINDy to incorporate inputs and outputs control and implementation for model predictive control [18, 116, 128]. Furthermore, SINDy extends to the identification of dynamics with rational function nonlinearities [119], integral terms [129],

and highly corrupt and incomplete data [18, 130].

A general Pareto front analysis can also be applied to SINDy to assess a combinatorially large set of potential dynamical models, discover the underlying sparse governing equations, and identify models with hidden variables using delay coordinates [131]. The optimal prediction model can be automatically identified for SINDy by assessing candidate models and determining the one with the lowest prediction error. SINDy accomplishes this task by developing a grid of λ_{SINDy} values, applying each cut-off to the data, and enumerating all resulting SINDy models to the prediction models. The method then uses new random initial conditions to generate 100 $\mathbf{X}_{val}$ validation data before comparing the predictions $\hat{\mathbf{X}}_{val}$ to the optimal state matrix model corresponding to the minimum relative corrected AIC_c [67]. However, SINDy with AIC will fail without sufficient data and when a low SNR masks the sampling of the dynamics [67].

Another variant of SINDy involves integrating ensembling methods to improve predictions and reliability (Ensemble-SINDy) [132]. In this approach, Ensemble-SINDy performs bragging, or robust bootstrapping, using bootstrap sampling to discover a range of models that are aggregated by taking the median of the estimates [132]. The identified ensemble of model coefficients can then be used to compute probability density functions, which form a posterior distribution $\mathbb{P}(\mathbf{B}|\mathbf{X})$ [132]. Furthermore, Ensemble-SINDy applies an additional threshold to the inclusion probability and removes terms that do not surpass that value [132].

Although SINDy has developed many avenues for data-driven system identification, it still has several drawbacks. One significant issue is the potential for degeneracy in the algorithm when dealing with complex functional forms. This occurs when certain mathematical models or terms in the underlying equations, due to their complexity or similarity, produce indistinguishable or nearly identical outcomes in the system's behaviour. This ambiguity complicates the process of accurately identifying the system's underlying dynamics and necessitates a hierarchical approach to the identification process. Here, terms of varying orders are systematically integrated until the SINDy algorithm reaches a point of convergence or, conversely, fails to find a solution [46]. Furthermore, SINDy requires a sufficiently large time series

and is not fully automated since it requires users to determine the hyperparameters for the numerical derivative from noisy data [46, 67].

2.10 Research Gaps

System identification remains a promising yet challenging area of study, characterised by certain limitations in current methodologies. Rooted in the initial question posed in Chapter 1 on the exploration of discovering dynamical systems, this chapter has provided a comprehensive overview of existing statistical methods, with a special emphasis on the SINDy framework [46]. SINDy aims to provide interpretable forms for the data's governing equations but relies on libraries of candidate functions and, therefore, has difficulty expressing complex dynamics. While other methods have also used the Gaussian process to identify underlying dynamics [47, 64, 67, 129], they face the same overarching issue. Even the extensions and new methods introduced in this thesis automatically perform system identification but still ultimately encounter this problem. It must be clarified that regression-based algorithms will always struggle to determine the true governing terms if they do not exist in the candidate library, which should be intuitive since these methods rely on the predictors in the design matrix to develop their predictions. Although this pitfall may occur, the results show that the novel methods outlined here can improve current identification techniques by adding more nonlinear terms to the design matrix. With more predictors available for selection, regression-based algorithms are more likely to discover the governing equations that describe the behaviour of the complex system in data, and this thesis develops adaptive and reliable frameworks for accomplishing this goal.

To address the advancement of model discovery by integrating sparse regression with statistical inference, Chapter 3 introduces a novel algorithm that employs sparse regression with statistical inference amidst noisy data conditions. Furthermore, each chapter employs sparse regression methods with smoothing techniques in its numerical differentiation schemes [46, 47], approximates weights of the terms with error bars using a thresholded Bayesian approach [68], detects highly corrupted

measurements [130], reduces errors using integral terms [129], and handles measurement noise by representing data as a Gaussian process [48]. These methods enable engineers to account for uncertainty in the measurements by allowing for nonzero covariance in their representation. However, recent studies indicate that identification with probabilistic methods faces the challenge of extracting a system that shifts from a learnable, low-noise phase to a stage where the observation noise is too high for any approach to learn the correct model [133]. This thesis's previous techniques and novel methods suffer from this fundamental issue and the problem of automatically developing a numerical derivative from data. Therefore, an optimal Savitzky-Golay filtering algorithm is developed here to provide the best predictions for the noisy signal from data and further advance automation within these novel frameworks.

The research question posed for Chapter 4, which emphasizes refining sparse identification through clustering-based algorithms, underscores a lack of rigorous approaches in evaluating algorithmic capabilities. Historically, researchers have used specific initial conditions for each system to perform identification on that data set [46, 47, 64, 67]. This thesis aims to demonstrate a systematic analysis that enables engineers to examine a given method better and trust its ability to discover a particular system from data. The nuanced approach employed here uses random initial condition bounds to expand each system, building separate matrix grids increasing in observations and SNR values. In performing this procedure, researchers can better understand how well their method uncovers their data's governing equations by calculating a success rate for each system.

In Chapter 5, potential enhancements within the ARGOS framework by employing Bayesian methods to improve efficiency. At its core, Bayesian regression offers a principled approach to handling uncertainties, allowing for probabilistic interpretations of system parameters and dynamics. Through this integration, the methodology gains the ability to capture the inherent uncertainties in the data and improves the computational efficiency of the framework. This amalgamation, therefore, promises a more robust and informed system identification, positioning ARGOS-BI as a potent tool for researchers navigating complex dynamical systems.

By far, the most important pitfall confronted here is the problem of the lack of automation within system identification. Researchers have endeavoured to build techniques to circumvent this issue. Still, either fail to automate the calculation of the numerical derivative, require previous knowledge of the underlying system, or prefer users manually implement a sparsity-promoting tuning parameter to determine the optimal estimates [46, 63, 64, 67, 104, 106]. The approaches described in the following chapters leverage these flaws and develop several novel techniques for automatically identifying dynamical systems from data, ensuring that researchers can trust that the resulting models are optimal in describing the equations for their data.

2.11 Summary

- Regression problems represent data estimates $\mathbf{y} \in \mathbb{R}^n$ as a linear combination of columns from the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. To determine the predicted formula for $\mathbf{y}$, linear regression is applied to estimate the coefficient vector $\beta \in \mathbb{R}^p$ as [46, 50–52],

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta},$$

- Sparse regression imposes a shrinkage penalty in the form of the ℓ_q norms to reduce the coefficients' values and the RSS of a given model for $q > 0$ [51],

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}.$$

  Sparse regression is commonly employed to perform automatic variable selection.

- K–fold cross-validation provides a different model assessment procedure that expands the data by creating K partitions of roughly equal-sized subsets [51, p.241]. The K estimates of prediction error are combined after performing this method for $k = 1, 2, \ldots, K$, using a different fold for the validation set each time [51, p.242].

$$\mathrm{CV}_{(K)} = \frac{1}{K} \sum_{i=1}^{K} \mathrm{MSE}_i,$$

  With K–fold cross-validation, researchers review a set of candidate models of varying flexibility and identify the λ model with the minimum test error. When performing cross-validation, researchers aim to determine how well they can expect a given statistical learning model to perform on independent data [50, p.183].

- Bootstrap sampling creates a normal distribution of samples with replacement. Developing bootstrap confidence intervals provides uncertainty measures to identify a model consisting of variables whose confidence intervals do not cross zero and whose point estimates fall within their confidence intervals. These intervals are calculated using the lower bound estimate, $CI_{\text{lo}} = [B\alpha/2]$, is the integer part of $B\alpha/2$, while the upper bound estimate is $CI_{\text{up}} = B - CI_{\text{lo}} + 1$ [90, p.24].

- K-means clustering is a popular unsupervised learning algorithm used for partitioning a set of n observations into k clusters. The algorithm separates a data set into k groups of similar observations based on the mean distance of the points within each group of clusters. Furthermore, K-means is a popular technique for clustering due to its simplicity, scalability, and efficiency.

- Dynamical systems are mathematical models that describe the temporal evolution of a process using ODEs and PDEs. Here, dynamical systems are represented as [18, 46]

$$\frac{d}{dt}x(t) = \dot{x}(t) = f(x(t)).$$

  System identification attempts to determine the underlying equations that describe dynamical systems.

- Savitzky-Golay filtering develops a generalised moving average technique to smooth a noisy signal before differentiation. Initially, the technique determines coefficients by performing an unweighted linear regression with a polynomial to smooth the data without distorting the signal tendency and maintaining the signal's shape [103, 125]. When using these parameters, the window length must be odd to centre around a given point, and the polynomial order must be less than the window length.

CHAPTER 3

Automatically Identifying Dynamical Systems from Data

Discovering nonlinear differential equations that describe system dynamics from empirical data is a fundamental challenge in contemporary science. This chapter introduces a novel approach, ARGOS, which combines denoising methods and sparse regression to construct bootstrap confidence intervals for the identification of dynamical systems. The efficacy of ARGOS in automating model discovery is demonstrated through a systematic analysis, showing that the method consistently outperforms the established SINDy with AIC [67] in identifying ordinary differential equations contaminated by significant levels of noise, particularly in three dimensions. By accurately discovering dynamical systems automatically, our methodology has the potential to impact the understanding of complex systems, especially in fields where data are abundant, but developing mathematical models demands considerable effort.

3.1 Methods

3.1.1 Automatic regression for governing equations

ARGOS aims to automatically discover interpretable models that describe the dynamics of a system by integrating machine learning with statistical inference. As illustrated in Figure 3.1, the algorithm comprises several key phases, including data smoothing, numerical approximation of derivatives, sparse regression, and bootstrap sampling for model selection to solve the system in Eq. (2.51).

ARGOS uses the Savitzky-Golay filter to approximate the derivative numerically, a popular tool for signal denoising [103]. ARGOS determines the optimal filtering parameters by setting polynomial order $o = 4$ and building a grid of window lengths l [124]. For each column of the noisy state matrix $\tilde{\mathbf{X}}$, ARGOS identifies the optimal l^* corresponding to the minimum MSE between $\tilde{\mathbf{x}}_j$ and its smoothed signal $\mathbf{x}_j$. With the optimal parameters, ARGOS uses the Savitzky-Golay filter to derive $\dot{\mathbf{x}}_j$ and consolidate the smoothed $\mathbf{X}$ and $\dot{\mathbf{X}}$ before constructing $\boldsymbol{\Theta}(\mathbf{X})$ with monomials up to the d-th degree (see Algorithm 1).

Algorithm 1 Automatic Savitzky-Golay Filter

Input: $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times m}$, dt.
Output: Savitzky-Golay optimally smoothed $\mathbf{X}$ and $\dot{\mathbf{X}}$.

1: determine lower and upper bounds of (odd) window length l:
$l_{min} = 13$,
$l_{max} = \max\left(13,\ \min\left(n - (n-1) \bmod 2, 101\right)\right)$;
2: build $L = (l_{min}, \ldots, l_{max})$;
▷ v = degree of derivative
3: **for** $j = 1, \ldots, m$ **do**
$l^* = \text{argmin}_L \left\| SG(\tilde{\mathbf{x}}_j, o = 4, l = L, v = 0, dt) - \tilde{\mathbf{x}}_j \right\|_2^2$
$\mathbf{x}_j = SG(\tilde{\mathbf{x}}_j, o = 4, l = l^*, v = 0, dt)$;
$\dot{\mathbf{x}}_j = SG(\tilde{\mathbf{x}}_j, o = 4, l = l^*, v = 1, dt)$;
4: **end for**
5: consolidate $\mathbf{X}, \dot{\mathbf{X}} \in \mathbb{R}^{n \times m}$ with each $\mathbf{x}_j$ and $\dot{\mathbf{x}}_j$, respectively.

ARGOS then performs identification by extracting each column of $\dot{\mathbf{X}}$ and $\mathbf{B}$ from Eq. (2.51):

$$\dot{\mathbf{x}}_j = \boldsymbol{\Theta}(\mathbf{X})\beta_j + \epsilon_j, \qquad j = 1, \ldots, m, \tag{3.1}$$

and determine a prediction model for each $\dot{\mathbf{x}}_j$ by applying either the lasso [76] or the

adaptive lasso [81] throughout the model selection process (see Algorithm 2). Both algorithms add the ℓ_1 penalty to the OLS regression estimate to shrink coefficients to zero, enabling ARGOS to select the nonzero terms for parameter and model inference. Since the weighted penalty in the adaptive lasso can be interpreted as an approximation of the ℓ_p penalties with $p = 1 - \nu$, ARGOS fixes $\nu = 1$, which allows the adaptive lasso to develop a soft-threshold approximation to the ℓ_0 penalty [52]. Furthermore, this approach provides an alternative to the hard thresholding applied by the recent SINDy algorithm, which requires a choice of the cut-off hyperparameter [46].

After identifying an initial sparse regression estimate of β_j in Eq. (3.1), the design matrix is trimmed to include only monomial terms up to the highest-order variable with a nonzero coefficient in the estimate. Using the updated design matrix, the previous sparse regression algorithm is reapplied, and a grid of thresholds is employed to develop a subset of models, with each model containing only coefficients whose absolute values exceed their respective thresholds. Next, OLS is performed on the selected variables of each subset to calculate unbiased coefficients and determine the point estimates from the regression model with the minimum Bayesian information criterion (BIC) [86]. As a final step, bootstrap sampling is employed with this sparse regression process with the trimmed design matrix to obtain 2000 sample estimates [89]. Finally, 95% bootstrap confidence intervals are constructed using these sample estimates and a final model is identified consisting of variables whose confidence intervals do not include zero and whose point estimates lie within their respective intervals.

3.1.2 Algorithm implementation

Since multicollinearity in the data is a concern, ARGOS applies ridge regression to obtain stable pilot estimates for Eq. (2.33). Moreover, ARGOS uses `glmnet` [80] to solve Eqs. (2.27) and (2.33) and produce a default λ grid before applying 10-fold cross-validation to determine the optimal initial tuning parameter λ_0^* [80]. After identifying λ_0^*, ARGOS creates a refined grid with 100 points corresponding to $[\lambda_0^*/10, 1.1 \cdot \lambda_0^*]$. The novel method then solves Eq. (2.33) by imposing the updated

Algorithm 2 Automatic Regression for Governing Equations (ARGOS)

Input: $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\dot{\mathbf{x}}_j \in \mathbb{R}^n$, d, $\alpha = 0.05$.

▷ STEP ONE: Initial design matrix

1: $p^{(0)} = \binom{m+d}{d}$;

2: create $\boldsymbol{\Theta}^{(0)}(\mathbf{X}) \in \mathbb{R}^{n \times p^{(0)}}$ with basis functions up to order d of the columns of $\mathbf{X}$;

▷ STEP TWO: Trim design matrix

▷ Variable selection with the lasso or adaptive lasso

▷ λ^* : Optimal λ from 10-fold cross-validation

▷ lasso: $w = 1$

▷ adaptive lasso: $w =$ ridge regression coefficients

3: $\hat{\beta}^{(0)} = \underset{\beta}{\operatorname{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(0)}(\mathbf{X})\beta \right\|_2^2 + \lambda^* \sum_{k=1}^{p^{(0)}} w_k |\beta_k|,$

4: extract $\boldsymbol{\Theta}^{(1)}(\mathbf{X})$ to contain columns of $\boldsymbol{\Theta}^{(0)}(\mathbf{X})$ up to the largest order of the selected variables in $\hat{\beta}^{(0)}$;

▷ STEP THREE: Final point estimates

▷ Repeat sparse regression algorithm from STEP TWO

5: $p^{(1)} = \binom{m+d^{(1)}}{d^{(1)}}$;

6: $\hat{\beta}^{(1)} = \underset{\beta}{\operatorname{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}(\mathbf{X})\beta \right\|_2^2 + \lambda^* \sum_{k=1}^{p^{(1)}} w_k |\beta_k|,$

▷ Apply threshold values

7: $\eta = [10^{-8}, 10^{-7}, \ldots, 10^{1}]$;

8: **for** $i = 1, \ldots,$ **card**(η) **do**

▷ Ordinary least squares regression (OLS) estimate after variable selection

$$\hat{\beta}^{\mathrm{OLS}}[i] = \underset{\beta_{\mathcal{K}_i}}{\operatorname{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}_{\mathcal{K}_i}(\mathbf{X})\beta_{\mathcal{K}_i} \right\|_2^2 \text{ where } \mathcal{K}_i = \{k : |\hat{\beta}^{(1)}_k| \geq \eta_i\},$$

$$\mathrm{BIC}_i = \mathrm{BIC}(\hat{\beta}^{\mathrm{OLS}}[i])$$

9: **end for**

10: $\hat{\beta} = \left\{ \hat{\beta}^{\mathrm{OLS}}[i] \middle| i : \operatorname{argmin}(\mathrm{BIC}) \right\}$

▷ STEP FOUR: Bootstrap estimates for confidence intervals

▷ $B = 2000$ bootstrap samples

11: bootstrap Statements 6 - 10 to approximate the confidence interval bounds: $CI_{\mathrm{lo}} = [B\alpha/2]$ and $CI_{\mathrm{up}} = B - CI_{\mathrm{lo}} + 1$;

12: construct bootstrap confidence intervals for $\hat{\beta}$: $\hat{\beta}_k \in \left[\hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{lo}}\}}, \hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{up}}\}} \right]$, and $0 < \hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{lo}}\}}$ or $0 > \hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{up}}\}}$

grid on `glmnet` and determining the model corresponding to the optimal λ^* that most accurately predicts $\dot{\mathbf{x}}_j$.

When applying ARGOS, the method uses $\eta = 10^{-8}, 10^{-7}, \ldots, 10^{1}$ to threshold the sparse regression coefficients before performing OLS on each subset $\mathcal{K} = \{k : |\hat{\beta}_k| \geq \eta\}$ of selected variables, determining an unbiased estimate for β [51]. ARGOS then calculates the BIC for each η regression model and selects the final model corresponding to the one with the minimum value [86].

The number of bootstrap sample estimates B must be large enough to develop robust confidence intervals for variable selection [89]. Therefore, ARGOS collects $B = 2000$ bootstrap sample estimates and sorts them by $\hat{\beta}_k^{\mathrm{OLS}\{1\}} \leq \hat{\beta}_k^{\mathrm{OLS}\{2\}} \leq \cdots \leq \hat{\beta}_k^{\mathrm{OLS}\{B\}}$. ARGOS then uses the 100(1 - α)% accuracy measure, where α denotes the significance level of each variable, to calculate the integer part of $B\alpha/2$ and develop estimates of the lower and upper bounds: $CI_{\mathrm{lo}} = [B\alpha/2]$ and $CI_{\mathrm{up}} = B - CI_{\mathrm{lo}} + 1$. Finally, ARGOS implements these calculations to develop confidence intervals $\left[\hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{lo}}\}}, \hat{\beta}_k^{\mathrm{OLS}\{CI_{\mathrm{up}}\}}\right]$ from the sample distribution [90].

To develop a fair comparison, Algorithm 1 was employed to filter the signal and differentiate the derivative numerically for SINDy with AIC.

3.2 Results

3.2.1 Building the data sets and tests

The results section examines the impact of limited data and noise on the performance of the system identification algorithms by conducting two sets of experiments. Central to the approach in this analysis is the use of a distribution of random initial conditions. By leveraging this strategy, the efficiency of ARGOS was evaluated with random data, reflecting its potential in real-world settings marked by inherent unpredictability and variability.

The first test keeps SNR = 49 dB constant and increases the number of observations n for each ODE system, using a grid of matrices with 100 random initial conditions. This enables the analysis to evaluate the identification performance of

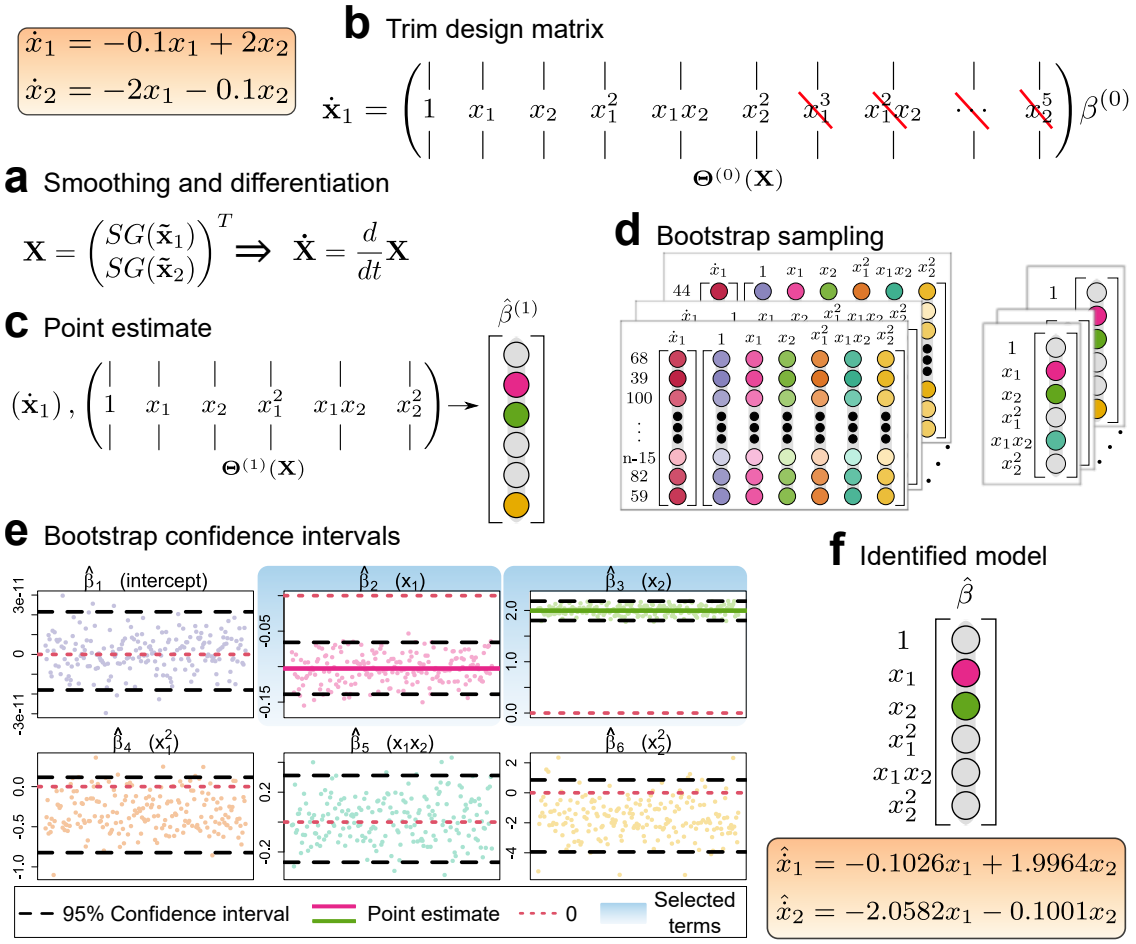

Figure 3.1: **Automatic regression for governing equations framework.** This example demonstrates the process of identifying the $\dot{x}_1$ equation of a two-dimensional damped oscillator with linear dynamics. (**a**) The algorithm initially smooths the state $\tilde{\mathbf{x}}_1$ and calculates the derivative $\dot{\mathbf{x}}_1$ vector. (**b**) The design matrix $\boldsymbol{\Theta}^{(0)}(\mathbf{X})$ is then constructed, containing the state-space variables and their interaction terms up to monomial degree $d = 5$, before trimming the matrix to only include terms up to the highest-order monomial degree of nonzero terms in each column of the estimate $\hat{\beta}^{(0)}$ (in this example, terms up to $d = 2$ are identified). Using $\boldsymbol{\Theta}^{(1)}(\mathbf{X})$, (**c**) sparse regression is again performed with the previously used algorithm (lasso or adaptive lasso), applying OLS on the subset of selected variables, and determining the final $\hat{\beta}^{(1)}$ point estimates. (**d**) Bootstrap sampling is subsequently employed to obtain 2000 sample estimates and (**e**) develop 95% bootstrap confidence intervals to (**f**) identify the $\hat{\beta}$ prediction model by selecting the coefficients whose confidence intervals do not include zero.

each algorithm when working with limited data. For each ODE other than the Lorenz system, the analysis uses temporal grids starting with $t_{\text{initial}} = 0$ and a varying t_{final} between 1 ($n = 10^2$) and 1000 ($n = 10^5$) with a time step $\Delta t = 0.01$. However, for the Lorenz equations, the analysis uses $\Delta t = 0.001$ to convert each maximum value of n to t_{final} between 0.1 ($n = 10^2$) and 100 ($n = 10^5$) [46].

The systematic analysis applies a similar process to observe the effect of noise on system identification. In this setting, the SNR in the grids varies by corrupting the state matrix with zero-mean Gaussian noise $\mathbf{Z} \sim \mathcal{N}(0, \sigma_{\mathbf{Z}}^2)$ (see Section 2.7 for review). Keeping n constant, the analysis again uses 100 random initial conditions before generating a grid of $\tilde{\mathbf{X}}$ matrices by adding noise to the system such that $\text{SNR} = 1, 4, \ldots, 61$ dB with $\Delta\text{SNR} = 3$ dB, including a noiseless system ($\text{SNR} = \infty$).

For both the increasing n and increasing SNR grids, the design matrix $\boldsymbol{\Theta}^{(0)}(\mathbf{X})$ of each system was built with monomial functions up to $d = 5$ of the smoothed columns of $\mathbf{X}$ [46]. ARGOS and SINDy with AIC then perform system identification with the matrix grids, enabling the calculation of their success rates as the probability of extracting the true underlying terms and observing their most frequently selected variables. This metric allows for quantitatively measuring the performance of each algorithm across different dynamical systems, as well as different SNR and n values (see Tables 3.1 and 3.2).

Furthermore, the results in Section 3.4 reserve 80% of the data for training and 20% to test each prediction model and apply trapezoidal integration to approximate the predictions of each algorithm for the corresponding system. The analysis then compares the predictions of each algorithm with the original $\tilde{\mathbf{X}}$ and uses the Frobenius norm to calculate the test set MSE:

$$\text{MSE} = \frac{1}{n_{\text{test}}} \cdot \|\mathbf{A}\|_F^2 = \frac{1}{n_{\text{test}}} \cdot \text{tr}(\mathbf{A}'\mathbf{A}), \tag{3.2}$$

where $\mathbf{A} = \hat{\mathbf{X}} - \mathbf{X}$ for $\text{SNR} \leq \infty$ [51, 52].

3.2.2 Assessing ARGOS systematically

To evaluate the effectiveness of our approach, several well-known ODEs were expanded using 100 random initial conditions, emulating real-world settings where one cannot select these initial values. Data sets were then generated with varying time series lengths n and SNRs (see Section 3.1.2) before a success rate metric was introduced, defined as the proportion of instances where an algorithm identified the correct terms of the governing equations from a given dynamical system. This metric quantitatively measures the performance of an algorithm across different dynamical systems, as well as different SNR and n values (see Tables 3.1 and 3.2). Figure 3.2 highlights success rates exceeding 80%, demonstrating that the proposed method consistently outperformed SINDy with AIC in identifying the underlying system from the data. ARGOS accurately represented linear systems with less than 800 data points and medium SNRs, underscoring its ability to handle straightforward dynamics. Notably, even with only moderately-sized data sets or medium SNRs, the approach successfully identified three out of five of the two-dimensional ODEs using the lasso, showcasing the effectiveness of integrating classic statistical learning algorithms within our framework. The adaptive lasso was able to identify the non-linear ODEs in three dimensions with higher accuracy than the other algorithms tested. These results suggest that the adaptive lasso is suitable for identifying non-linear ODEs in higher dimensional systems.

The systematic analysis, presented in Figure 3.2, emphasised the efficacy of the proposed approach as n and SNR increased. The importance of data quality and quantity is further supported by Figure 3.3, which illustrates the frequency at which ARGOS identified each term in the design matrix across different values of n and SNR. The boxes in the figure delineate regions where each algorithm achieved model discovery above 80% for the Lorenz system, providing insights into the selected terms contributing to the success and failure of each method across different settings. When faced with limited observations and low signal quality, ARGOS identified overly sparse models that failed to represent the governing dynamics accurately, while SINDy with AIC selected erroneous terms, struggling to obtain a parsimonious representation of the underlying equations. Figure 3.3 also illustrates

Table 3.1: Minimum number of observations (n) needed for each method to obtain 80% accuracy in identifying governing equations of dynamical systems. Top-performing algorithms are in red, and three-dimensional systems have a shaded background. See Appendix A for further details on governing equations.

System	Algorithm	n
Two-dimensional linear	ARGOS-Lasso	$10^{2.6}$ (399)
	ARGOS-Adaptive Lasso	$10^{2.6}$ (399)
	SINDy with AIC	$10^{3.3}$ (1996)
Three-dimensional linear	ARGOS-Lasso	$10^{2.9}$ (795)
	ARGOS-Adaptive Lasso	$10^{3.2}$ (1585)
	SINDy with AIC	NA
Two-dimensional cubic	ARGOS-Lasso	$10^{3.2}$ (1585)
	SINDy with AIC	$10^{3.3}$ (1996)
	ARGOS-Adaptive Lasso	$10^{4.1}$ (12590)
Lotka-Volterra	ARGOS-Adaptive Lasso	$10^{3.2}$ (1585)
	SINDy with AIC	$10^{3.2}$ (1585)
	ARGOS-Lasso	$10^{3.3}$ (1996)
Rossler	ARGOS-Adaptive Lasso	$10^{2.9}$ (795)
	ARGOS-Lasso	$10^{3.2}$ (1585)
	SINDy with AIC	$10^{3.2}$ (1585)
Lorenz	ARGOS-Adaptive Lasso	$10^{3.8}$ (6310)
	ARGOS-Lasso	$10^{3.9}$ (7944)
	SINDy with AIC	NA
Van der Pol	ARGOS-Adaptive Lasso	$10^{2.9}$ (795)
	SINDy with AIC	$10^{2.9}$ (795)
	ARGOS-Lasso	$10^{3.0}$ (1000)
Duffing	ARGOS-Lasso	$10^{2.6}$ (399)
	SINDy with AIC	$10^{2.9}$ (795)
	ARGOS-Adaptive Lasso	$10^{3.0}$ (1000)

Table 3.2: Minimum signal-to-noise ratio (SNR) tolerated by each method to achieve 80% accuracy in identifying the governing equations of the dynamical systems. Top-performing algorithms are in red, and three-dimensional systems have a shaded background. See Appendix A for further details on governing equations.

System	Algorithm	SNR
Two-dimensional linear	ARGOS-Lasso	25
	ARGOS-Adaptive Lasso	25
	SINDy with AIC	37
Three-dimensional linear	ARGOS-Lasso	31
	ARGOS-Adaptive Lasso	40
	SINDy with AIC	∞
Two-dimensional cubic	ARGOS-Lasso	43
	SINDy with AIC	46
	ARGOS-Adaptive Lasso	NA
Lotka-Volterra	ARGOS-Adaptive Lasso	16
	SINDy with AIC	22
	ARGOS-Lasso	28
Rossler	ARGOS-Adaptive Lasso	31
	ARGOS-Lasso	34
	SINDy with AIC	NA
Lorenz	ARGOS-Adaptive Lasso	46
	ARGOS-Lasso	55
	SINDy with AIC	∞
Van der Pol	SINDy with AIC	16
	ARGOS-Adaptive Lasso	19
	ARGOS-Lasso	25
Duffing	ARGOS-Lasso	28
	ARGOS-Adaptive Lasso	28
	SINDy with AIC	34

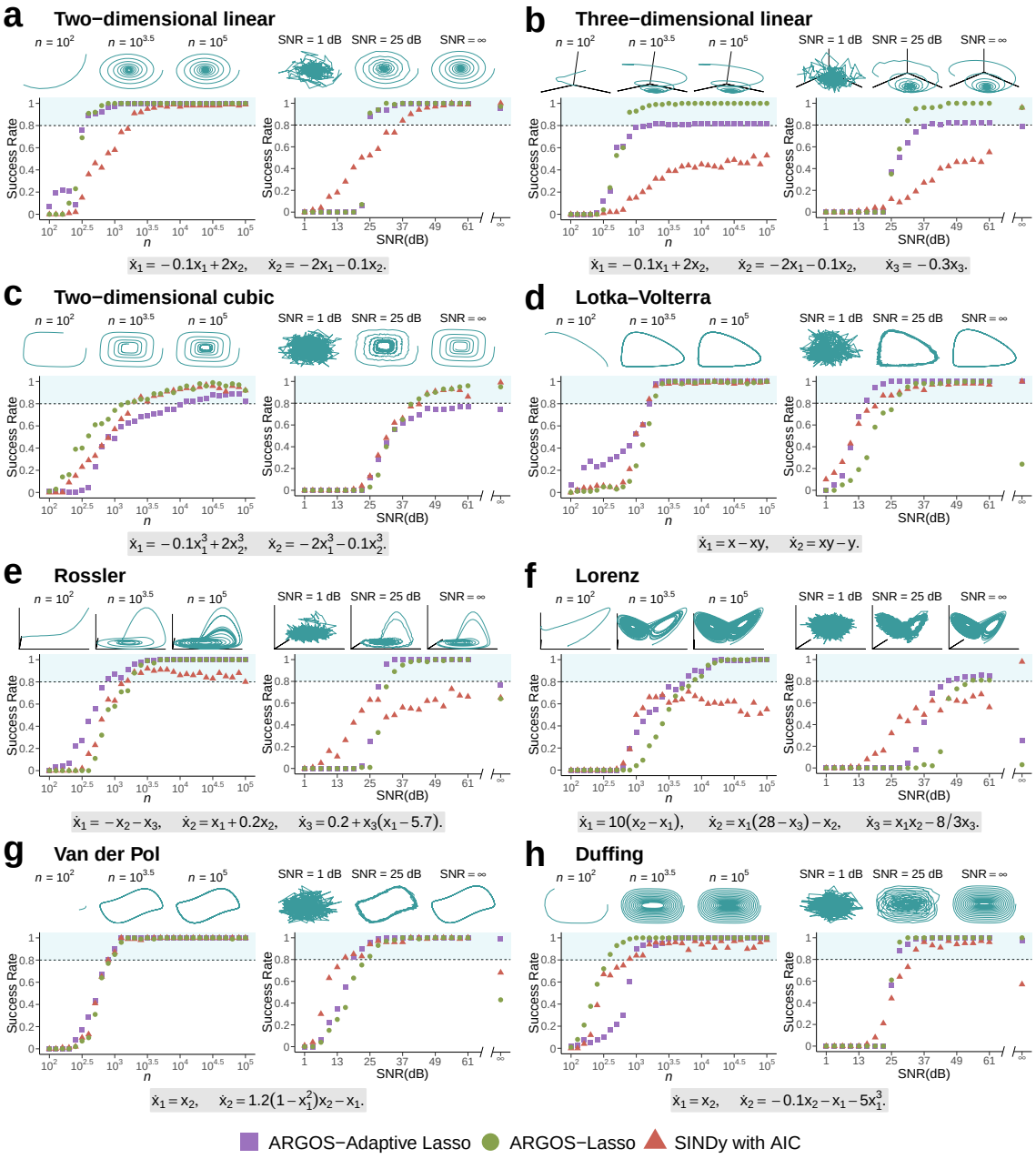

Figure 3.2: **Success rate of ARGOS versus SINDy with AIC for linear and nonlinear systems**. 100 random initial conditions were generated to examine the success rate of ARGOS and SINDy with AIC in correctly discovering each system at each value of n and SNR. (**a**)-(**b**) Linear systems. First-order nonlinear systems in two (**c**)-(**d**) and three (**e**-(**f**) dimensions. The time-series length n increases while holding SNR = 49 dB (left panels) and fixing $n = 5000$ when increasing the SNR (right panels). Shaded regions represent model discovery above 80%.

the decline in the proposed method's performance for deterministic systems, as it identified several ancillary terms for the Lorenz dynamics when SNR $= \infty$. The decrease in identification accuracy stemmed from the identified model's violation of the homoscedasticity assumption in linear regression, which occurs when residuals exhibit non-constant variance. Figure 3.4 demonstrates that the identified model from ARGOS did not satisfy this assumption when identifying the $\dot{x}_1$ equation of the Lorenz system. Consequently, the proposed approach selected additional terms to balance the variance among the model's residuals while sacrificing correct system discovery. As the noise in the system slightly increased, however, homoscedasticity in the residuals became more pronounced, enabling ARGOS to distinguish the equation's true underlying structure. Thus, ARGOS proved more practical in accurately identifying the correct terms of the governing equations when data contained low levels of noise in the signal, which is often the case in many real-world applications, as opposed to when dealing with noiseless systems.

The proposed method outperformed SINDy with AIC in identifying a range of ODEs, especially three-dimensional systems. One potential explanation for the lesser performance of SINDy with AIC is that multicollinearity in the design matrix often causes OLS to produce unstable coefficients. Due to the sensitivity of the estimated coefficients, small changes in the data can lead to fluctuations in their magnitude, making it difficult for the sparsity-promoting parameter to determine the correct model. As a result, the initial phase of the hard-thresholding procedure of SINDy with AIC inadvertently removed the true dynamic terms of the underlying system. Therefore, this model selection approach will likely face persistent challenges when discovering higher-dimensional systems that contain additional multicollinearity in the design matrix.

Figure 3.5 shows the computational time, measured in seconds, required for the proposed approach and SINDy with AIC to perform model discovery. While ARGOS demanded greater computational effort for the two-dimensional linear system than SINDy with AIC, it demonstrated better efficiency in identifying the Lorenz dynamics as n increased. The decrease in efficiency of SINDy with AIC can be attributed to its model selection process, which involves enumerating all potential

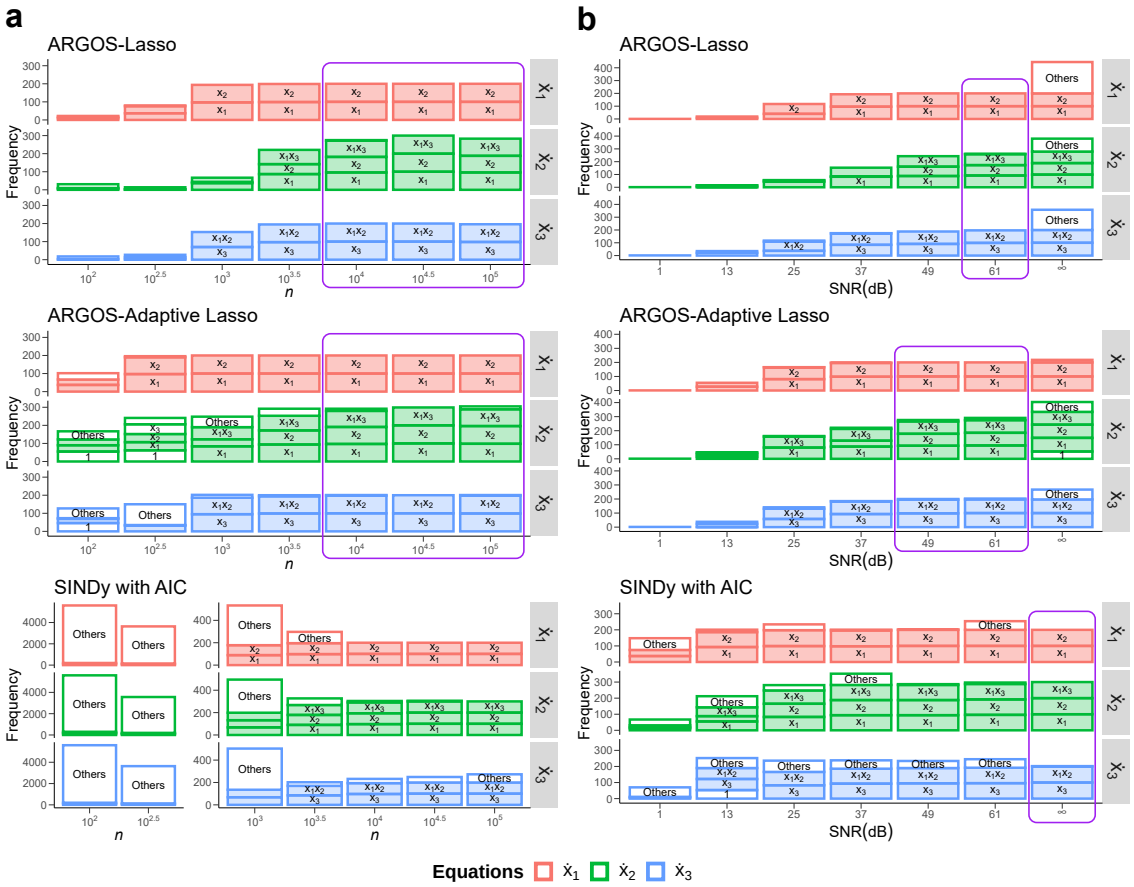

Figure 3.3: **Frequency of identified variables for the Lorenz system across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

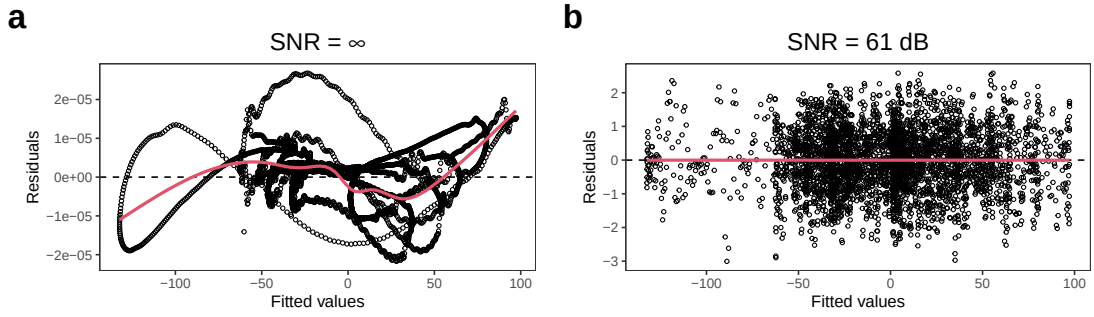

Figure 3.4: **Residuals vs fitted diagnostics for the ARGOS-Lasso identified model of the Lorenz $\dot{x}_1$ equation**. Comparison of residuals for the prediction models identified using the lasso with ARGOS for the Lorenz system's $\dot{x}_1$ equation when data are (**a**) noiseless and (**b**) contaminated by SNR = 61 dB.

prediction models—a procedure that becomes progressively more expensive with data in higher dimensions [67]. In contrast, the proposed approach displayed a similar increase in computational complexity as the time series expanded for both systems, suggesting that ARGOS was less affected by the growing data dimensionality than SINDy with AIC. Thus, ARGOS offers a more efficient alternative for identifying three-dimensional systems with increasing time series lengths.

Determining the grid of window lengths requires a constant that we cannot drop in the optimal Savitzky-Golay filtering algorithm. Therefore, the method's computational complexity can be described as $\mathcal{O}(nw)$. Furthermore, the computational complexity of the lasso and the adaptive lasso is $\mathcal{O}(p^3 + np^2)$ [81, 134]. The adaptive lasso performs this process twice, identifying ridge regression pilot estimates. Thus, engineers should be aware of the slight increase in efficiency. Finally, the complexity of ARGOS can be described by $\mathcal{O}(B(p^3 + p^2 n))$ since the bootstrap method applies the algorithm it is performing B times to develop as many sample estimates [89].

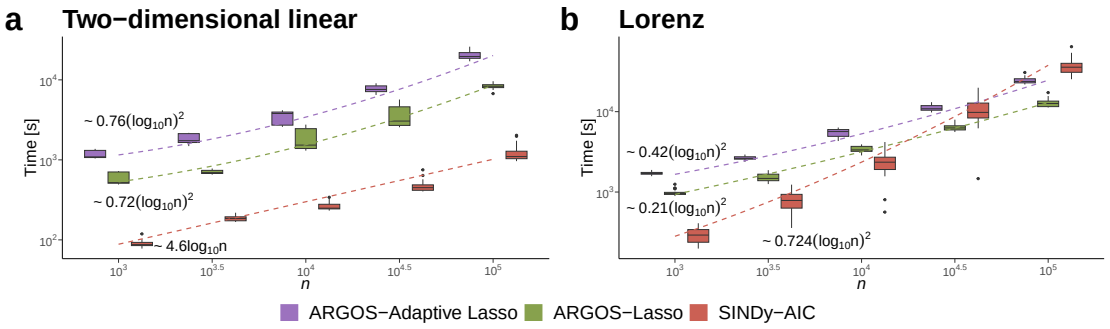

Figure 3.5: **Time-complexity (seconds) between ARGOS and SINDy with AIC**. Boxplots depict the computational time required for model discovery over 30 instances for (**a**) a two-dimensional damped oscillator with linear dynamics and (**b**) the Lorenz system. The black bar within each box represents the median computational time. Whiskers extending from each box show 1.5 times the interquartile range. Data points beyond the end of the whiskers are outlying points. Equations accompanying the dashed lines indicate the fitted mean computational time for each algorithm at various values of n.

3.3 Discussion

This chapter demonstrated a novel method, ARGOS, for extracting dynamical systems from scarce and noisy data without prior knowledge of governing equations.

The proposed method combines the Savitzky-Golay filter for signal denoising and differentiation with bootstrap sampling and sparse regression for confidence interval estimation, effectively addressing the inverse problem of inferring underlying dynamics from observational data through reliable variable section. By examining diverse trajectories, this chapter showcases the efficacy of the proposed algorithm in automating the discovery of mathematical models from data and consistently outperforming the established SINDy with AIC, especially when identifying systems in three dimensions, offering significant advances to researchers across disciplines.

Despite promising results, it is important to note several potential limitations of the proposed approach. First, although ARGOS effectively automates model discovery, it can only correctly represent the true governing equations if the active terms are present in the design matrix, a constraint inherent in regression-based identification procedures. Building on this point, the importance of data quantity and quality must be stressed, as identification accuracy improved with sufficient observations and moderate to high signal-to-noise ratios. The proposed method also performs better when data contains low levels of noise, as opposed to noiseless systems. Under noiseless conditions, the linear regression assumption of homoscedasticity is violated, and the method identifies spurious terms to develop a more constant variance among the residuals. However, this issue can be mitigated in the presence of a small amount of noise in the data, leading to a more constant variance in the residuals of the true model and enabling more accurate identification. Lastly, as the number of observations and data dimensionality increase, bootstrap sampling becomes computationally demanding, significantly prolonging the model selection process and limiting the proposed algorithm's applicability in real-time. Nonetheless, obtaining confidence intervals through bootstrap sampling serves as a reliable approach for ARGOS, allowing the method to eliminate superfluous terms and select the ones that best represent the underlying equations, ultimately leading to more accurate predictions of the system's dynamics.

In this information-rich era, data-driven methods for uncovering governing principles are increasingly crucial in scientific research. Combining statistical learning with model assessment techniques provides an effective and reliable process for iden-

tifying underlying equations and promoting automation. This chapter endorses an inference-based approach, emphasising the importance of thorough model evaluation for building confidence in discovering governing equations from data. Further developments in automatic system identification will hopefully further accelerate scientific discovery across various disciplines.

3.4 Additional Case Studies

3.4.1 Linear systems

Two-dimensional damped oscillator with linear dynamics

The two-dimensional damped oscillator with linear dynamics can be described as [46]

$$\begin{aligned} \dot{x}_1 &= -0.1x_1 + 2x_2, \\ \dot{x}_2 &= -2x_1 - 0.1x_2. \end{aligned} \tag{3.3}$$

The tests used a random uniform distribution containing 100 values between $[10^{-1}, 10^{3}]$ to develop $x_1(t)$ and $x_2(t)$. Figure 3.6 provides further details for the systems generated to demonstrate model discovery. Each trajectory rapidly converges to the true oscillator, allowing ARGOS to discover the governing equations.

Figure 3.7 illustrates the performance of ARGOS and SINDy with AIC in discovering the two-dimensional damped oscillator with linear dynamics. For conditions with limited data and low SNR, ARGOS identified overly sparse models and struggled to identify the underlying equations of the system. As the length of the time series n increased and the data became less contaminated with noise, the performance of ARGOS improved in extracting the true terms. Conversely, SINDy with AIC demonstrated a tendency to produce dense models, which contained numerous erroneous variables, particularly when n and SNR were low.

The MSE values in Figure 3.8(**a**) show that when each algorithm identified sparser models, their prediction error was reduced. However, Figure 3.8(**b**) illustrates that ARGOS began identifying models with fewer observations necessary

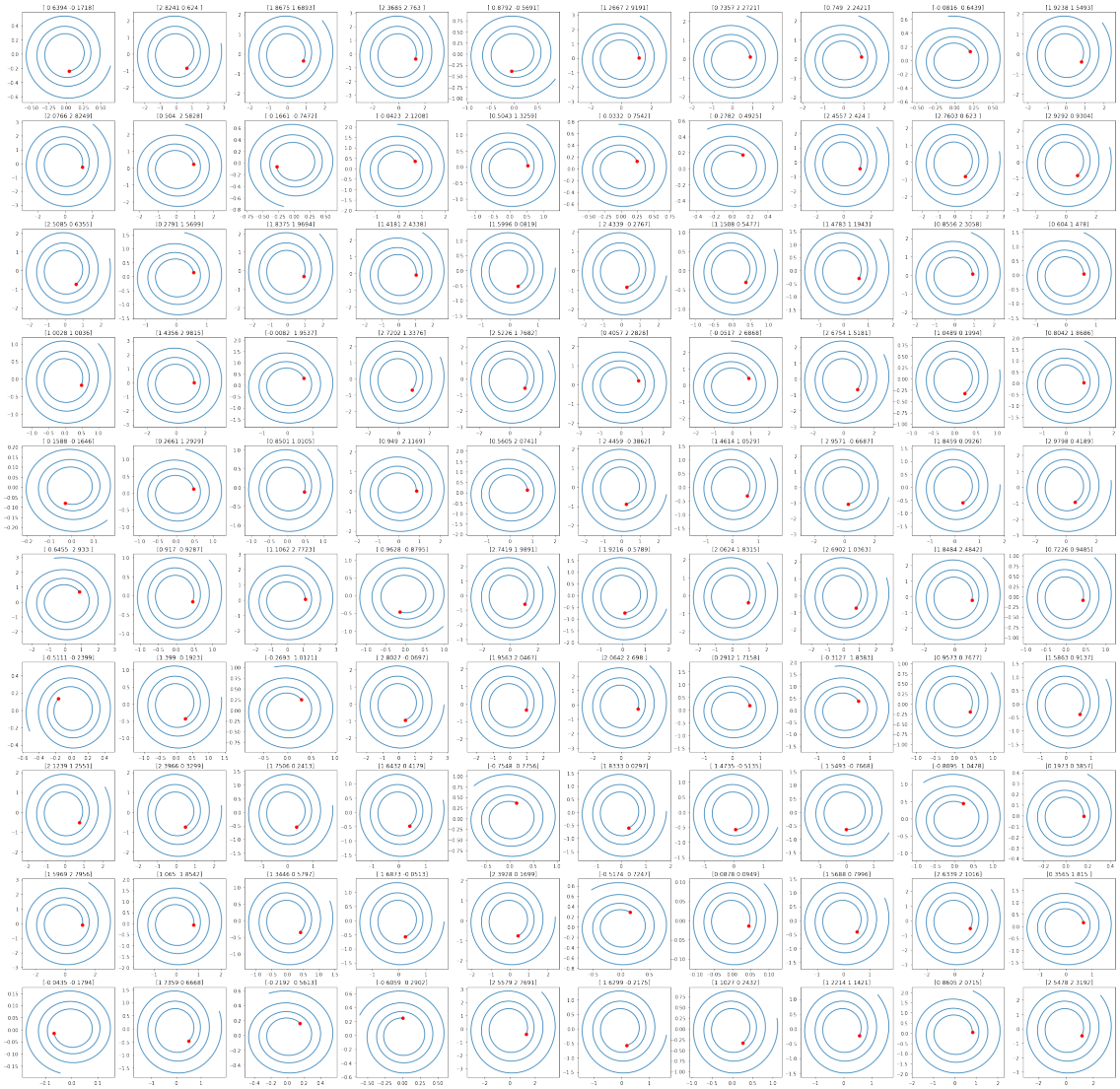

Figure 3.6: **100 Instances of a two-dimensional damped harmonic oscillator with linear dynamics.** The two-dimensional linear system was generated using a random uniform distribution containing 100 values between $[10^{-1}, 10^{3}]$ for the state variables $x_1(t)$ and $x_2(t)$.

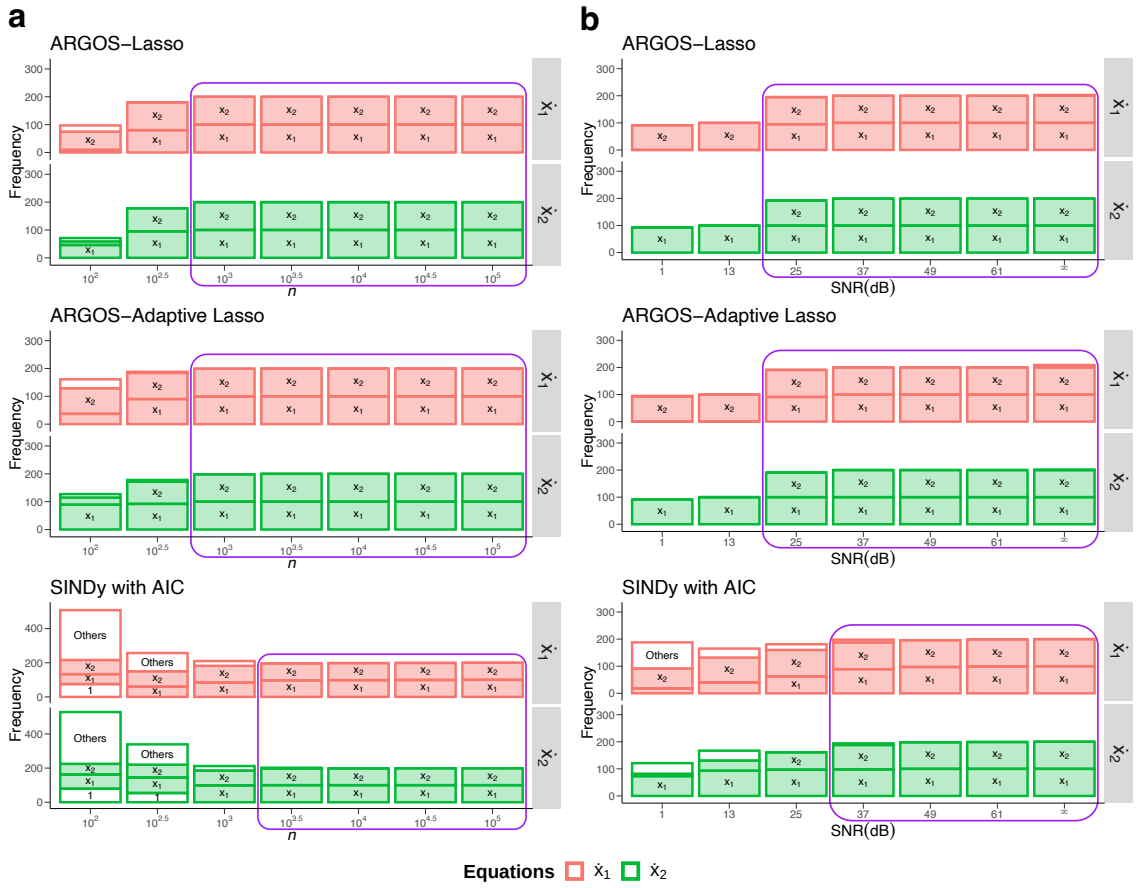

Figure 3.7: **Frequency of identified variables for the two-dimensional damped harmonic oscillator with linear dynamics across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

than SINDy with AIC, proving that it was capable of extracting the true equations and developing stronger predictions with limited data.

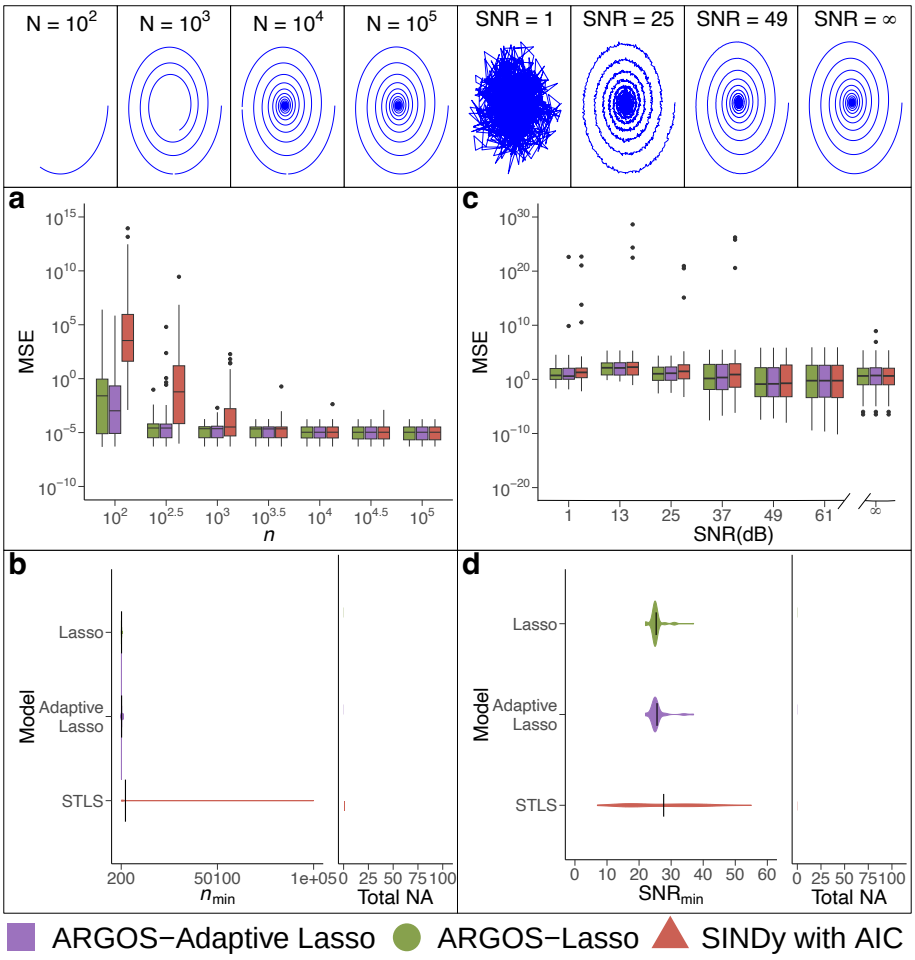

Figure 3.8: **MSE and minimum identification for the two-dimensional damped oscillator with linear dynamics.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

Figure 3.8(**c**) displays ARGOS consistently identifying models with a lower MSE than SINDy with AIC. Here, ARGOS showed that it could develop stronger predictions for the two-dimensional linear oscillator as noise increases in the system. Ultimately, Figure 3.8(**d**) exhibits ARGOS and SINDy with AIC handled noise similarly since they averaged a similar $\text{SNR}_{\min}$ until the system was no longer identifiable.

Three-dimensional linear system

The analysis evaluates a three-dimensional system [46]:

$$\begin{aligned} \dot{x}_1 &= -0.1x_1 + 2x_2, \\ \dot{x}_2 &= -2x_1 - 0.1x_2, \\ \dot{x}_3 &= -0.3x_3. \end{aligned} \tag{3.4}$$

In Figure 3.9, 100 random initial conditions are generated using a uniform distribution containing values between $[10^{-1}, 10^3]$ to expand the state variables $x_1(t)$, $x_2(t)$, and $x_3(t)$. Each trajectory rapidly converges to the true oscillator, allowing ARGOS to discover the governing equations.

Figure 3.10 shows that ARGOS more consistently discovered the three-dimensional system than SINDy with AIC, particularly when using the standard lasso algorithm. Again, the importance of data quality and quantity must be emphasised, as ARGOS identified overly sparse models that did not fully represent the system for low values of n and SNR. Additionally, SINDy with AIC consistently misidentified this linear system by selecting several erroneous terms.

In Figure 3.11(**a**), ARGOS again developed stronger predictions than SINDy with AIC as n increased in the matrix grid. These results show that, even as more data was obtained, ARGOS more consistently identified and predicted the three-dimensional linear system from data. Furthermore, on average, ARGOS began extracting the true equations of the underlying system around $n = 200$ (Figure 3.11(**b**)).

In Figure 3.11(**c**), ARGOS consistently identified models displaying a lower MSE than SINDy with AIC. ARGOS more frequently outperformed SINDy with AIC in extracting the system and developing predictions, providing a novel approach to system identification with noise-robust results. Finally, Figure 3.11(**d**) displays ARGOS better handling noise than SINDy with AIC since, on average, it continued to identify the system contaminated with larger SNR values. Therefore, by developing robust confidence intervals, ARGOS required a limited number of observations to uncover noisy linear systems from data.

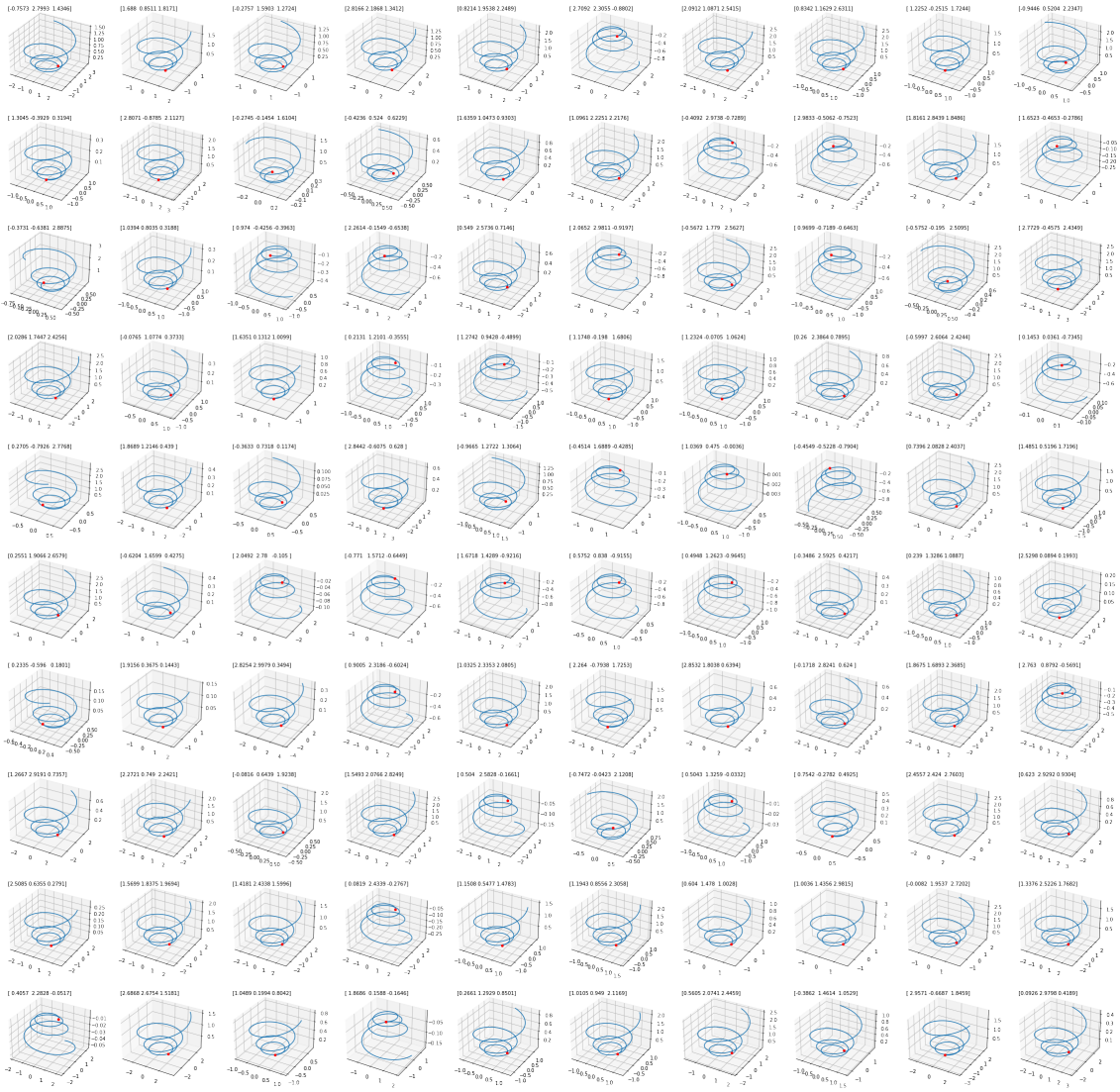

Figure 3.9: **100 Instances of a three-dimensional linear system.** The three-dimensional linear system was generated using a random uniform distribution containing 100 values between $[10^{-1}, 10^{3}]$ for the state variables $x_1(t)$, $x_2(t)$, and $x_3(t)$.

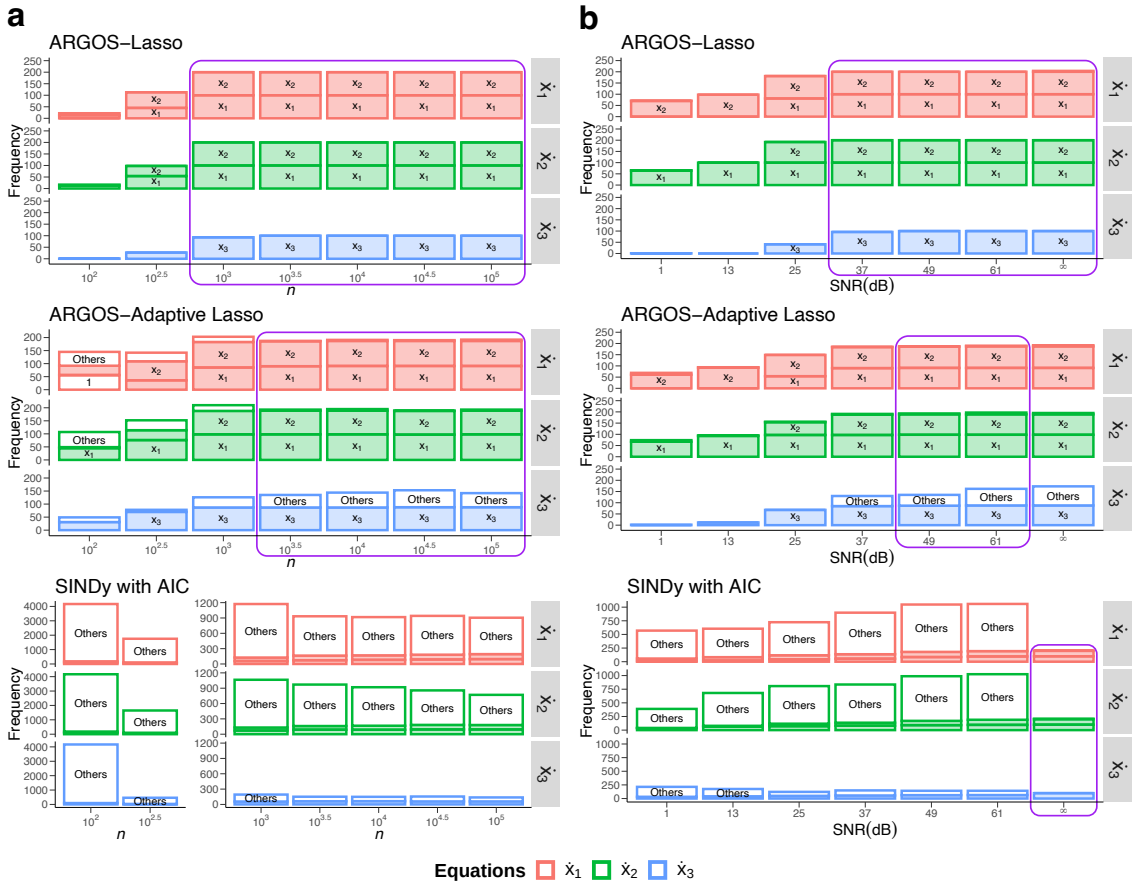

Figure 3.10: **Frequency of identified variables for the three-dimensional linear system across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR (n = 5000). Purple-bordered regions demarcate model discovery above 80%.

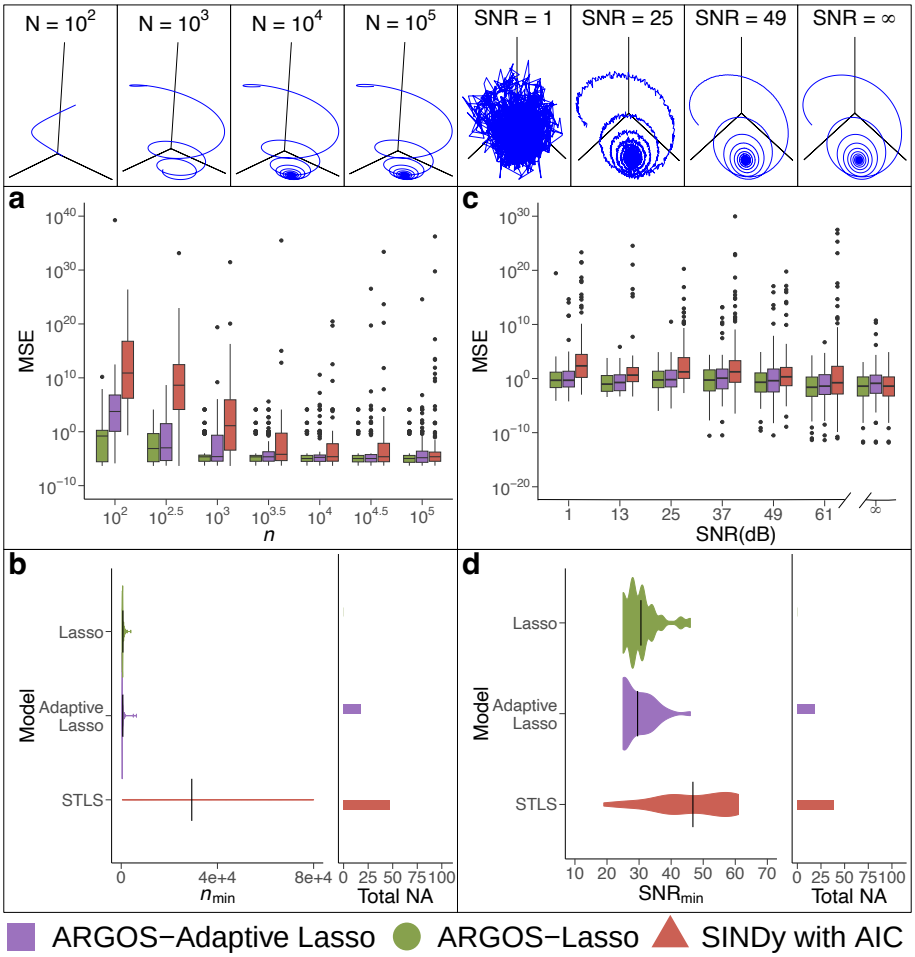

Figure 3.11: **MSE and minimum identification for the three-dimensional system.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

3.4.2 First-order nonlinear systems

Two-dimensional damped oscillator with cubic dynamics

The two-dimensional damped oscillator with cubic dynamics can be described as [46]

$$\begin{aligned} \dot{x}_1 &= -0.1x_1^3 + 2x_2^3, \\ \dot{x}_2 &= -2x_1^3 - 0.1x_2^3. \end{aligned} \tag{3.5}$$

Figure 3.12 shows the state variables $x_1(t)$ and $x_2(t)$ were developed using a random uniform distribution containing 100 values between $[-2, 2]$. These initial conditions ensure that each trajectory rapidly converges to the true oscillator with $n = 5000$ observations, enabling the system to be contained within the data.

Figure 3.13 demonstrates the effectiveness of the lasso algorithm in conjunction with ARGOS for identifying the two-dimensional damped harmonic oscillator with cubic dynamics. In this case, the proposed approach consistently outperformed SINDy with AIC. Thus, the lasso with ARGOS presents a powerful tool for discovering these cubic dynamics.

Figure 3.14(**a**) shows that ARGOS provided stronger predictions by consistently displaying lower MSE values than SINDy with AIC. Although the violin plots in Figure 3.14(**b**) display a wide range of $n_{\min}$ for identification, ARGOS-Lasso proves to be most successful in initially discovering the equations.

Figure 3.14(**c**) shows that, although each algorithm provided similar MSE values, ARGOS developed a narrower MSE distribution than SINDy with AIC, displaying less variability in its predictions. ARGOS consistently outperformed SINDy with AIC in extracting the system and developing predictions, providing a novel approach to system identification with noise-robust results. However, Figure 3.14(**d**) shows that SINDy with AIC displays similarly competitive results in handling noise, as it continued to identify the system contaminated with larger SNR values.

Lotka-Volterra system

Two first-order nonlinear differential equations describe the Lotka-Volterra equations, commonly utilised to depict the interaction dynamics between two species in

Figure 3.12: **100 Instances of a two-dimensional damped harmonic oscillator with cubic dynamics.** The two-dimensional cubic system was generated using a random uniform distribution containing 100 values between $[-2, 2]$ for the state variables $x_1(t)$ and $x_2(t)$.

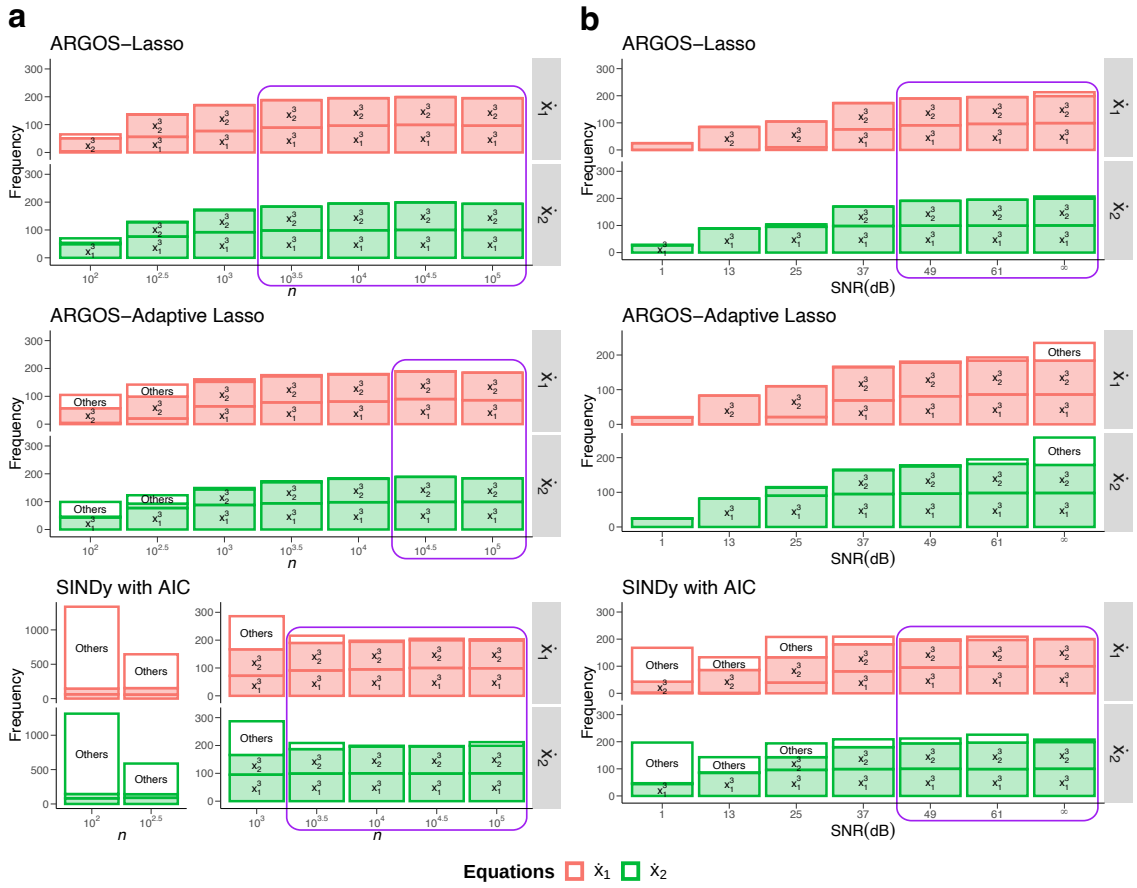

Figure 3.13: **Frequency of identified variables for the two-dimensional damped harmonic oscillator with cubic dynamics across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

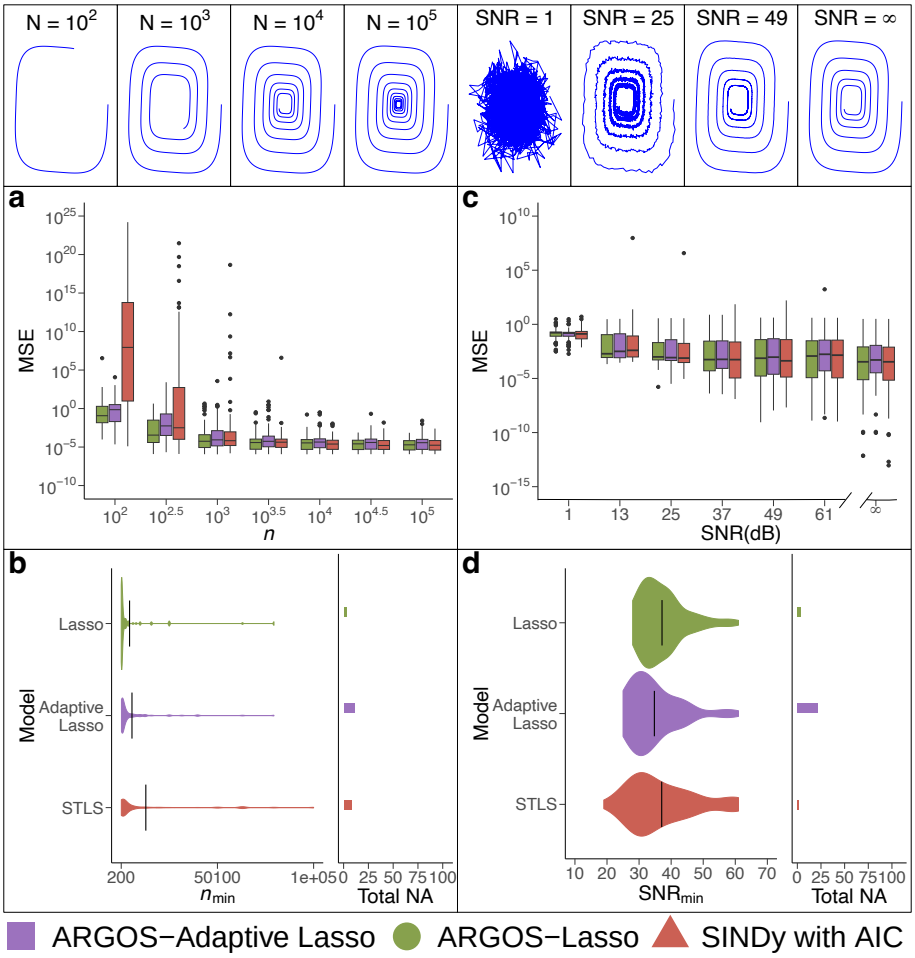

Figure 3.14: **MSE and minimum identification for the two-dimensional damped oscillator with cubic dynamics.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

biological systems, with one being the predator and the other the prey [135].

Identifying these equations accurately is essential as they are a fundamental model for studying predator-prey dynamics in biological systems. If researchers can accurately identify the Lotka-Volterra equation for a particular ecosystem, they can use this model to predict how environmental changes, such as climate change and human intervention, may affect the populations of different species. Researchers can then inform conservation efforts and other management practices to promote healthy ecosystems and seek sustainable resources. Furthermore, by accurately identifying the Lotka-Volterra equations, researchers can develop more effective control strategies for managing populations of invasive species, allowing us to develop targeted interventions to control the invasive population without disrupting the balance of the ecosystem. The predator-prey equations are represented as

$$\begin{aligned} \dot{x}_1 &= \alpha x_1 - \zeta x_1 x_2, \\ \dot{x}_2 &= \delta x_1 x_2 - \gamma x_2, \end{aligned} \tag{3.6}$$

where the prey birth rate $\alpha = 1$ and the predator death rate $\delta = -1$, and the interaction parameters $\zeta = -1$ and $\gamma = 1$ [136]. Since the population cannot start with a negative number, positive values were used for the 100 values of the random uniform distribution between $[1, 10]$ for $x_1(t)$ and $x_2(t)$ (see Figure 3.15). Surprisingly, the figure shows these systems have not fully developed into predator-prey dynamics. However, ARGOS and SINDy with AIC still identify the system with a high degree of accuracy.

Figure 3.16 illustrates the effectiveness of each method, particularly ARGOS-Adaptive Lasso, in discovering the governing equations of the Lotka-Volterra system. As n and SNR increased, ARGOS demonstrated the most consistent discovery of the true governing terms using the adaptive lasso within our framework. In contrast, SINDy with AIC tended to discover numerous erroneous terms, especially when data was limited.

Figure 3.17(a) shows that ARGOS provides stronger predictions with fewer observations than SINDy with AIC, corresponding to Figure 3.2(d) since ARGOS-Adaptive Lasso was the first method to begin identifying the system successfully.

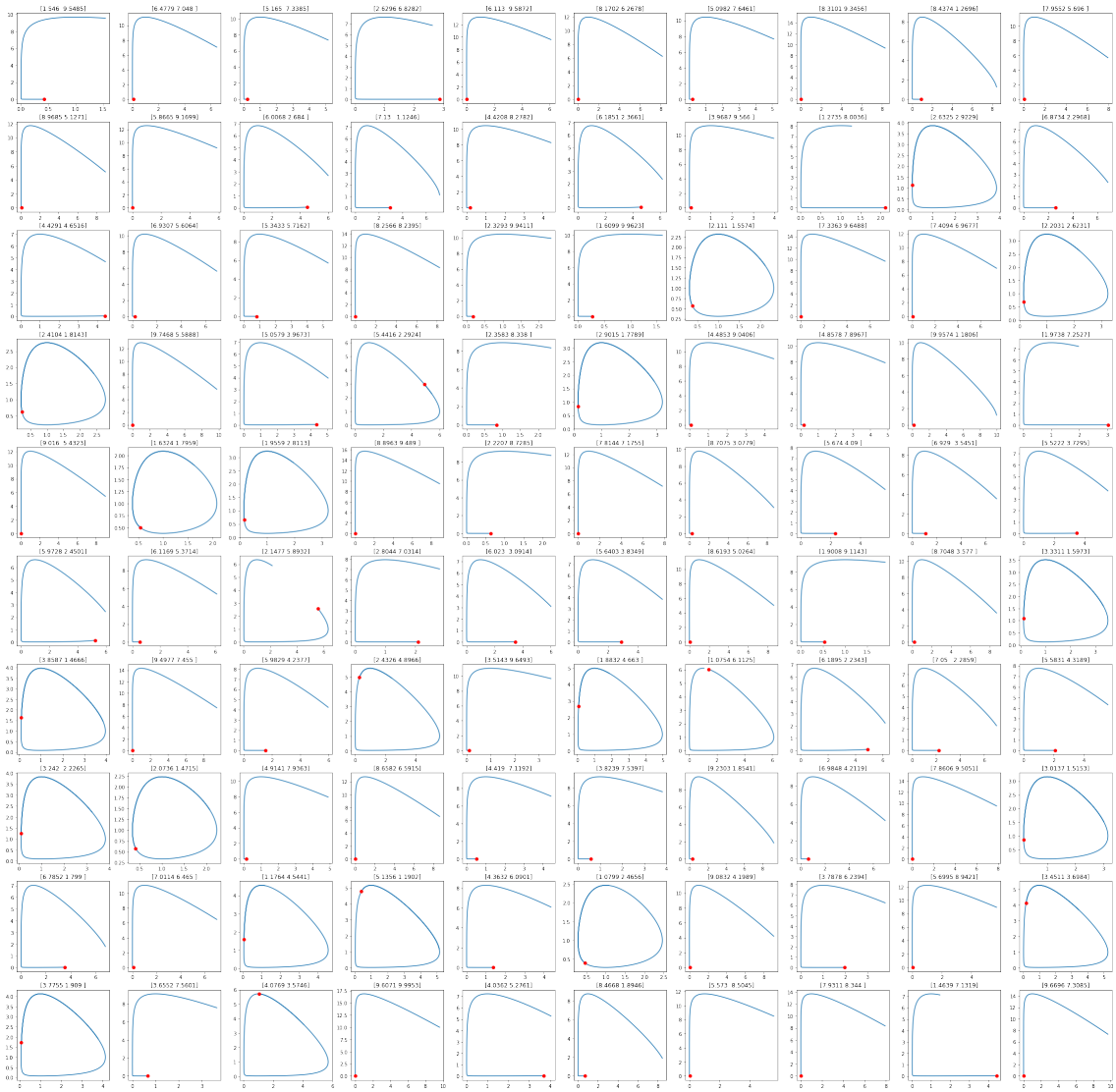

Figure 3.15: **100 Instances of the Lotka-Volterra system with initial conditions.** The Lotka-Volterra system was generated using a random uniform distribution containing 100 values between $[1, 10]$ for the state variables $x_1(t)$ and $x_2(t)$.

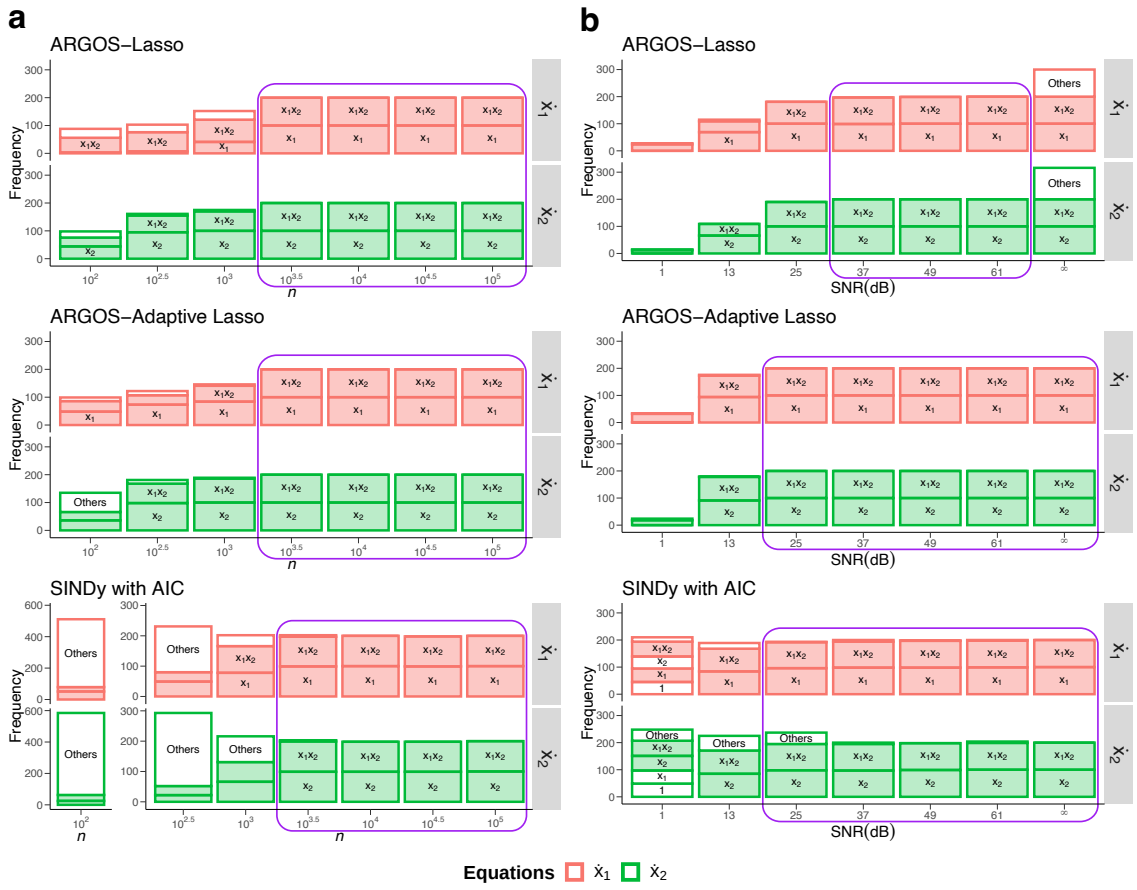

Figure 3.16: **Frequency of identified variables for the Lotka-Volterra system across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

However, as n continued to increase, each method began successfully discovering the governing equations and developing strong prediction models for the dynamics. Furthermore, each approach ultimately required very few observations to begin representing the underlying system from data, while ARGOS-Adaptive Lasso most consistently selected the correct model, only requiring $\bar{n}_{\min} \approx 158$ (Figure 3.17(**b**)).

In Figure 3.17(**c**), each algorithm provided similarly low MSE values, but ARGOS developed a smaller average MSE than SINDy with AIC as the system became more contaminated with noise. Moreover, Figure 3.14(**d**) displays the noise-robust results of each algorithm, while ARGOS on average, performed better with larger SNR values in the data.

Rossler system

The Rossler system is a three-dimensional chaotic system and is represented by the equations

$$\begin{aligned} \dot{x}_1 &= -x_2 - x_3, \\ \dot{x}_2 &= x_1 + a x_2, \\ \dot{x}_3 &= b + x_3(x_1 - c), \end{aligned} \tag{3.7}$$

where $a = 0.2$, $b = 0.2$, and $c = 5.7$ [130]. For $x_1(t)$, $x_2(t)$, and $x_3(t)$, the tests developed a random uniform distribution containing 100 values between $[-10, 10]$, $[-10, 10]$, and $[0, 20]$ (see Figure 3.18). These initial conditions ensure that each trajectory rapidly converges to the true system with $n = 5000$ observations, enabling the system to be contained within the data.

Figure 3.19 demonstrates the effectiveness of ARGOS in accurately representing the Rossler system, provided that sufficient data is available. The novel approach consistently identified the underlying dynamics and outperformed SINDy with AIC, which struggled to achieve high success rates. Specifically, SINDy with AIC failed to surpass 80% for any SNR value, highlighting its limitations in handling complex dynamics in three dimensions. In contrast, ARGOS proved more reliable, making it a superior choice for identifying the governing equations of chaotic systems under various conditions.

From Figure 3.20(**a**), each algorithm developed similarly low MSE distributions

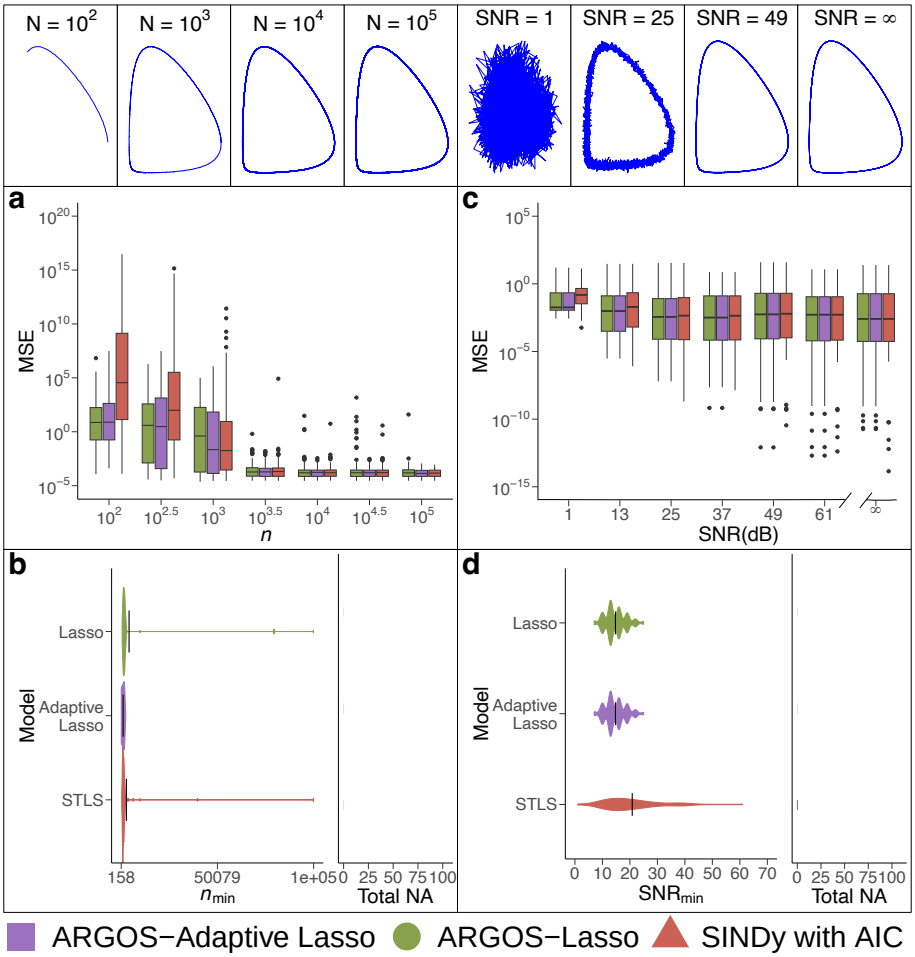

Figure 3.17: **MSE and minimum identification for the Lotka-Volterra system.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

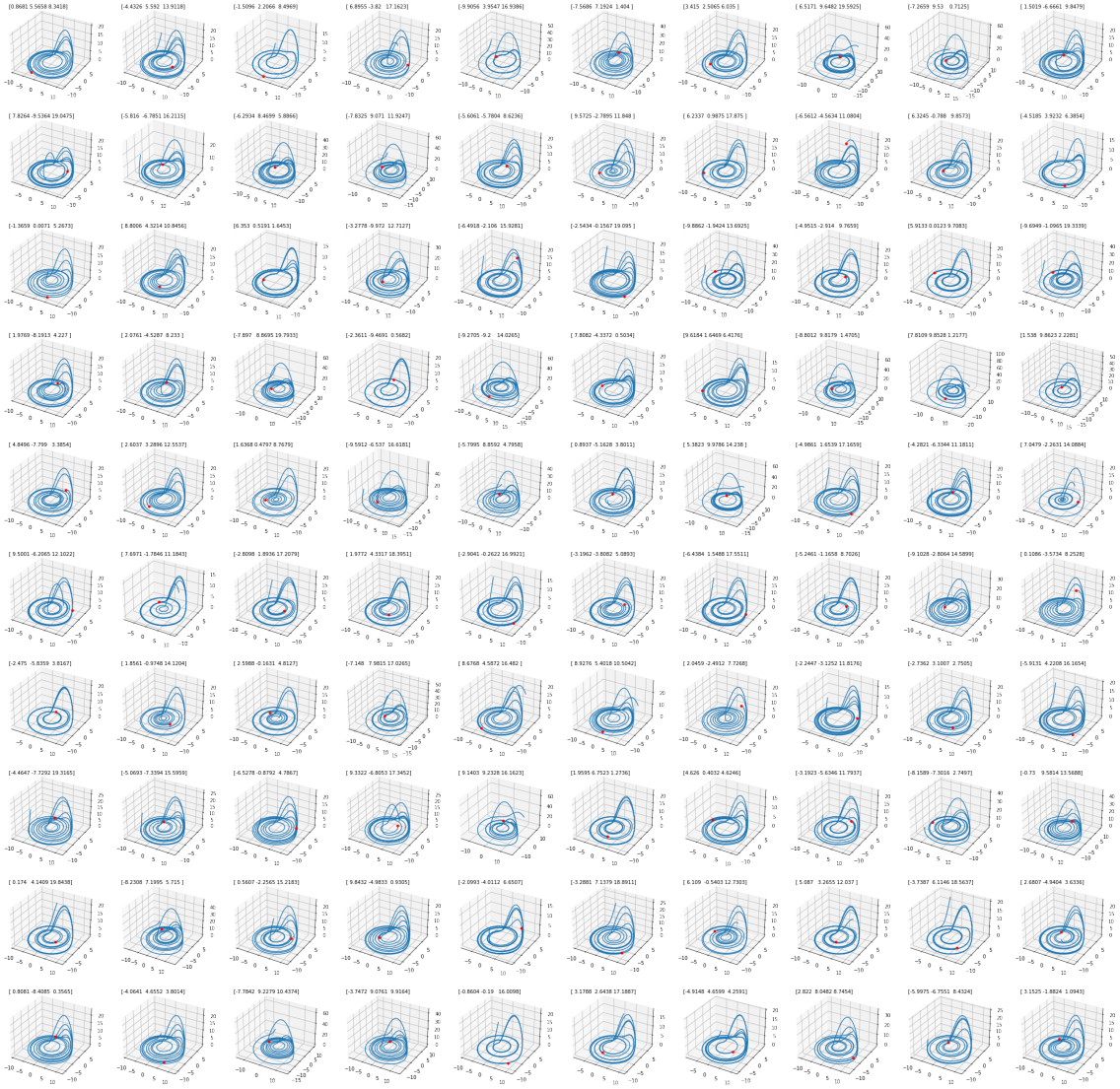

Figure 3.18: **100 Instances of the Rossler system.** Each system was generated using a random uniform distribution containing 100 values between $[-10, 10]$, $[-10, 10]$, and $[0, 20]$ for the state variables $x_1(t)$, $x_2(t)$, and $x_3(t)$.

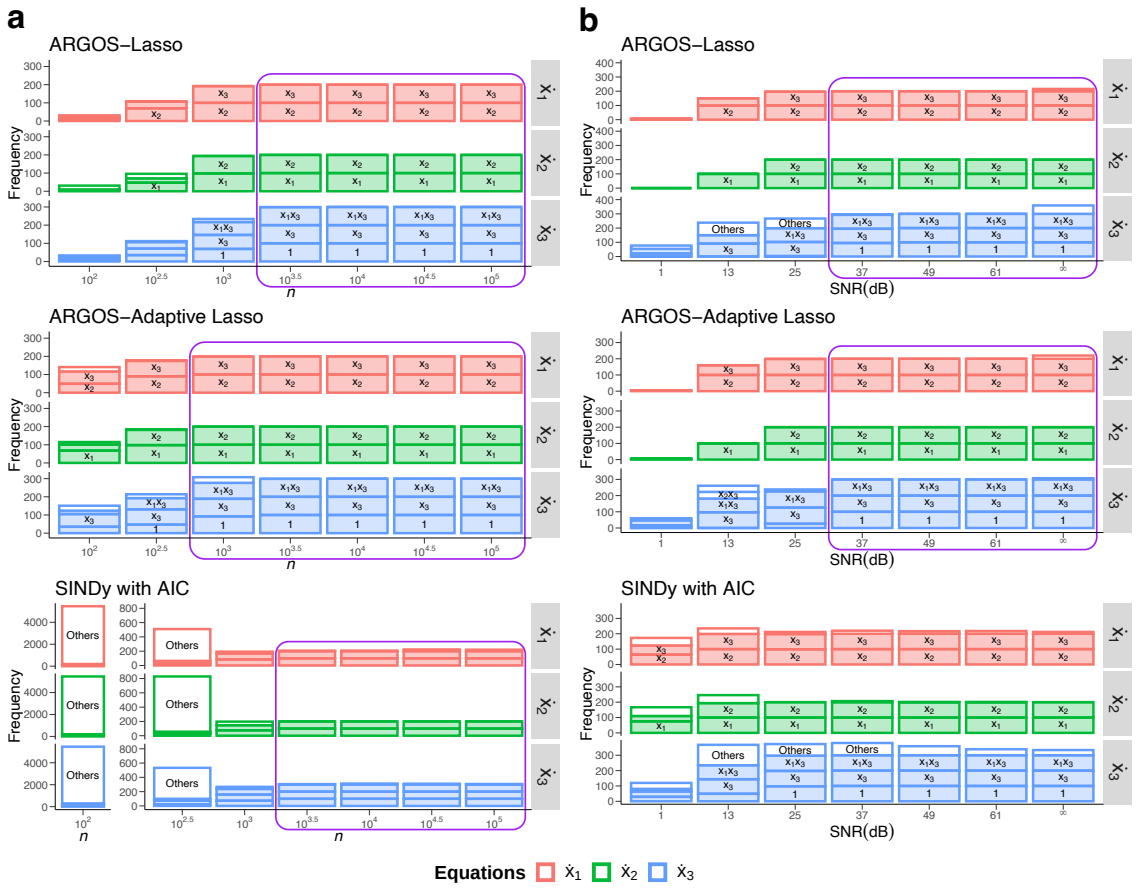

Figure 3.19: **Frequency of identified variables for the Rossler system across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR ($n = 5000$). Purple-bordered regions demarcate model discovery above 80%.

as n increased in the system. However, ARGOS again displayed a lower MSE distribution with fewer observations. Furthermore, in Figure 3.20(**b**), ARGOS required significantly fewer observations to begin identifying the system and continued to do so as the number of observations increased.

Figure 3.20(**c**) displays SINDy with AIC providing a lower MSE distribution than ARGOS for smaller values of SNR. As the SNR increased in the system, ARGOS began identifying the equations correctly, leading to a lower MSE distribution. Here, SINDy with AIC benefited from prior knowledge of the underlying system, which enabled the method to extract the true equations even with significant noise. Ultimately, however, ARGOS handled noise much better than SINDy with AIC since the method consistently identified the equations as the system became more contaminated with noise (Figure 3.20(**d**)).

Lorenz equations

The Lorenz chaotic system is a low-dimensional nonlinear structure originally a simple model for atmospheric convection [46]. Researchers model the Lorenz system using a set of ODEs:

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2, \\ \dot{x}_3 &= x_1 x_2 - \zeta x_3, \end{aligned} \tag{3.8}$$

with the values of the original parameters $\sigma = 10$, $\rho = 28$, and $\zeta = 8/3$ [46]. Figure 3.21 shows 100 instances of the system expanded by values in a random uniform distribution containing values between $[-15, 15]$, $[-15, 15]$, and $[10, 40]$. These values ensure the attractor is contained within the data, enabling a precise, systematic analysis. Section 3.2.2 provides more detail regarding each algorithm's identification of the Lorenz system.

In Figure 3.22(**a**), each algorithm provided similarly low MSE distributions as n increased in the system while ARGOS provided lower values around $n = 100$. Furthermore, Figure 3.22(**b**) shows that ARGOS required significantly fewer observations to begin identifying the system ($n \approx 631$) than SINDy with AIC ($n \approx 10^5$).

Figure 3.22(**c**) shows that SINDy with AIC provided a lower MSE distribution

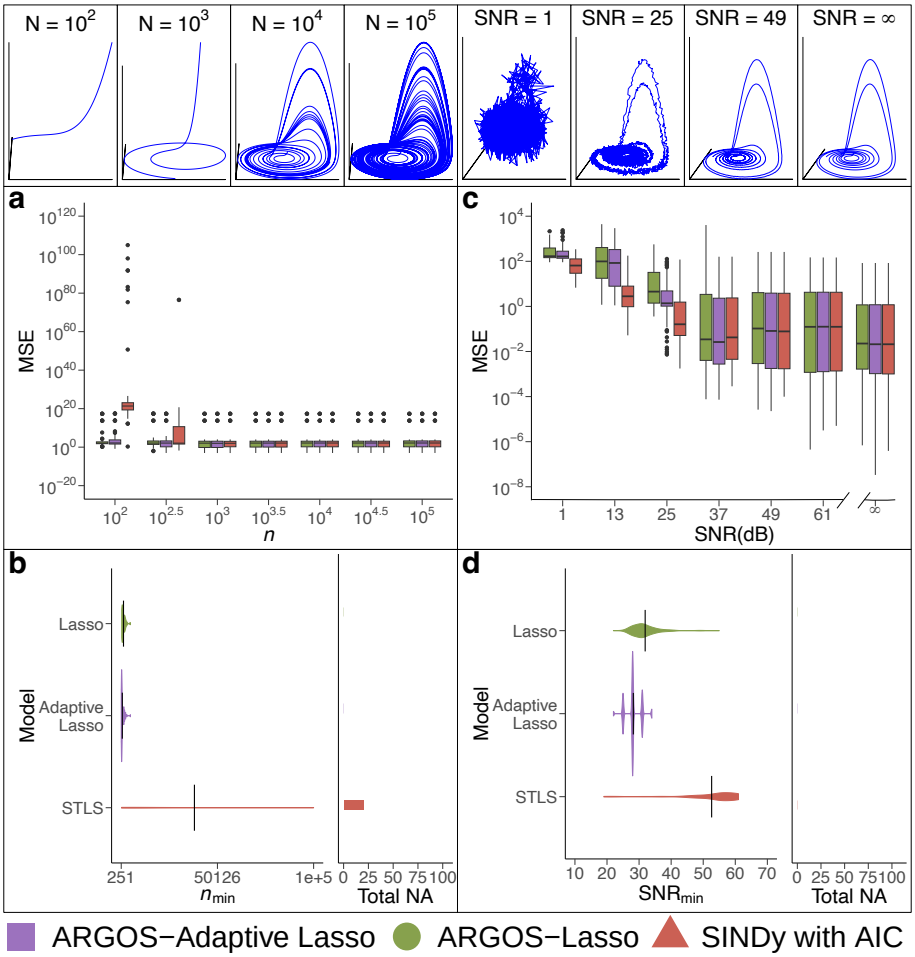

Figure 3.20: **MSE and minimum identification for the Rossler system.** The figure is vertically divided into two regions: keeping the SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\text{min}}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\text{min}}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

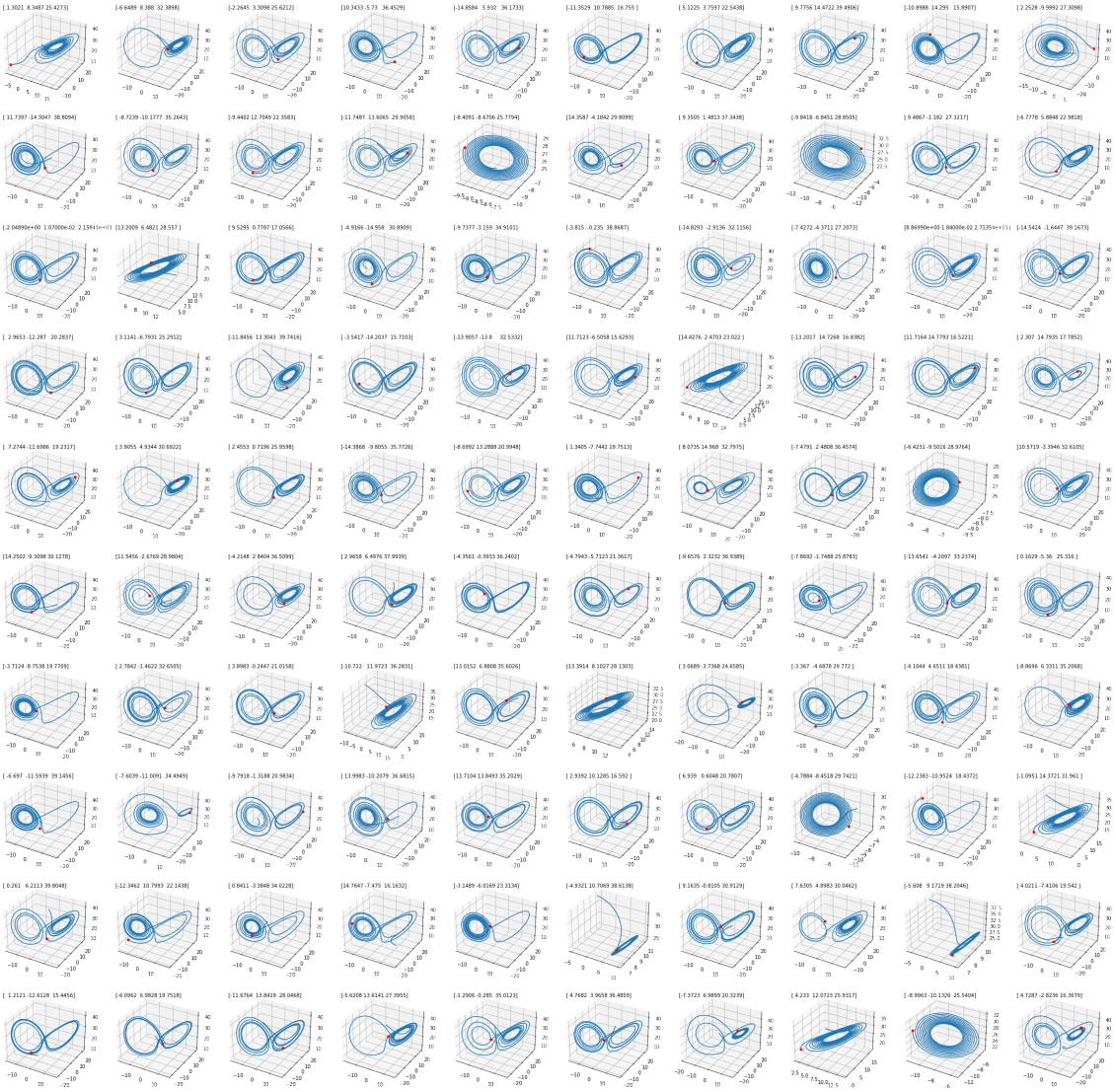

Figure 3.21: **100 Instances of the Lorenz system.** Each system was generated using a random uniform distribution containing 100 values between $x_1(t) \in [-15, 15]$, $x_2(t) \in [-15, 15]$, and $x_3(t) \in [10, 40]$.

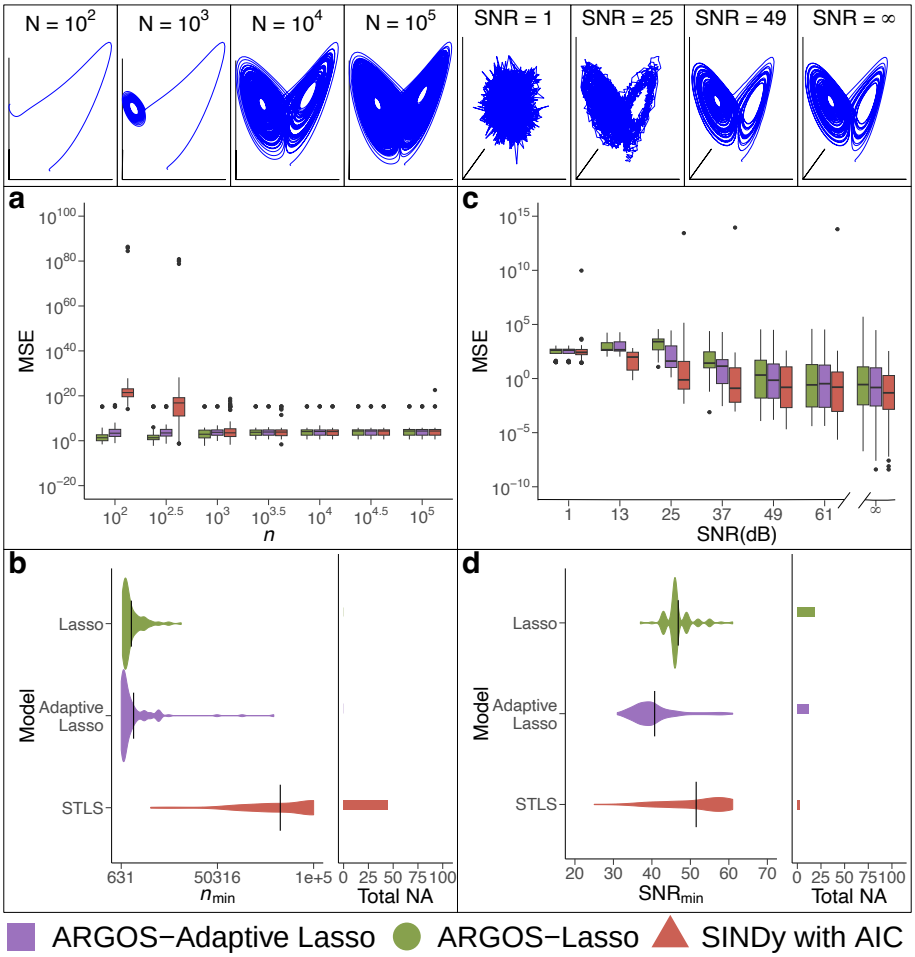

Figure 3.22: **MSE and minimum identification for the Lorenz system.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\text{min}}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\text{min}}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

than ARGOS for smaller values of SNR. ARGOS identification dropped at SNR = ∞, showing that its MSE distribution subsequently increased when compared to SINDy with AIC. Interestingly, ARGOS displayed more noise-robust results than SINDy with AIC since it extracted the system with higher SNR values and continued to do so as the system became less contaminated with noise (Figure 3.22(**d**)). If ARGOS successfully uncovered the system, it would continue to do so as SNR $\to \infty$.

3.4.3 Second-order nonlinear systems

Van der Pol oscillator

The canonical oscillator was initially proposed as a nonlinear circuit model with a triode tube in 1922 and is a mathematical model used to describe self-sustaining oscillations in physical and biological systems [64, 137]. As a result of the nonlinear damping in the $\dot{x}_2$ equation, the oscillator reaches a stable periodic state resulting in limit cycle behavior [64]. The Van der Pol oscillator is represented as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - x_1, \end{aligned} \tag{3.9}$$

where $\mu = 1.2$ controls the nonlinear damping level of the system [64]. In Figure 3.23, 100 random initial conditions are generated between $[-4, 4]$ for $x_1(t)$ and $x_2(t)$ to expand the Van der Pol oscillator. The figure shows that the oscillator is well represented in the data, which allows for a systematic comparison between ARGOS and SINDy with AIC.

Figure 3.24 presents a comparison between our approach and SINDy with AIC in discovering the governing equations for the Van der Pol oscillator. Initially, our method developed overly sparse models, while SINDy with AIC produced dense models for the underlying dynamics. However, as n and SNR increased, ARGOS demonstrated a marked improvement in accurately representing the behaviour of the oscillator.

In Figure 3.25(**a**), the sparse models identified by ARGOS provided a lower MSE distribution with fewer observations than the dense models extracted by SINDy with

Figure 3.23: **100 Instances of the Van der Pol oscillator.** The Van der Pol oscillator was generated using a random uniform distribution containing 100 values between $[-4, 4]$ for the state variables $x_1(t)$ and $x_2(t)$.

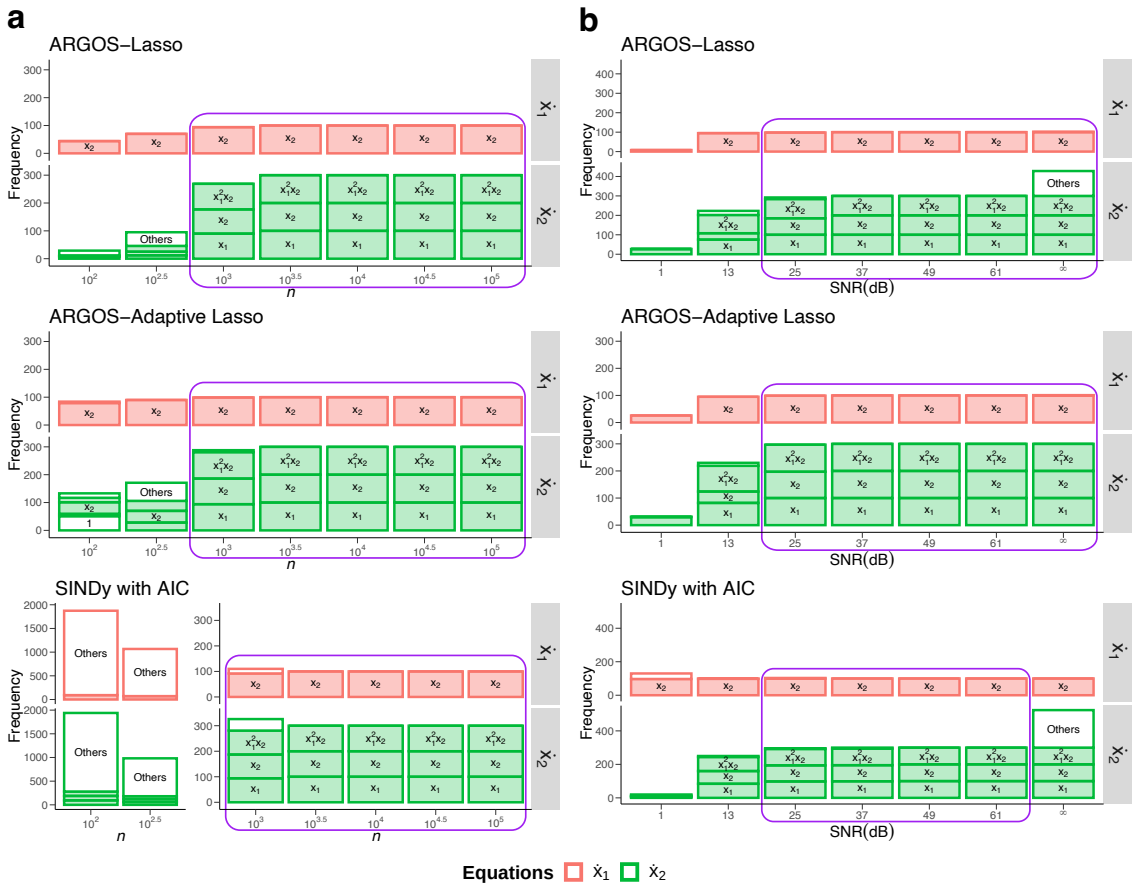

Figure 3.24: **Frequency of identified variables for the Van der Pol oscillator across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR (n = 5000). Purple-bordered regions demarcate model discovery above 80%.

AIC. However, as n increased, each algorithm provided similarly low distributions since they eventually uncovered the true equations. Furthermore, Figure 3.25(**b**) shows that each algorithm required very few observations to extract the true terms of the Van der Pol oscillator contaminated with limited noise.

Figure 3.25(**c**) again shows that each algorithm performed similarly well when uncovering prediction models representative of the canonical oscillator. Figure 3.25(**d**) further displays these results since ARGOS and SINDy with AIC develop similar $\overline{\text{SNR}}_{\text{min}}$ values for the system. Ultimately, each algorithm was competitive in extracting the Van der Pol oscillator.

Duffing oscillator

The Duffing oscillator provides an alternative cubic nonlinear system that can represent chaos and often models a spring-damper-mass system that contains a spring with a restoring force of $f(\zeta) = -\kappa\zeta - \epsilon\zeta^3$, where $\epsilon > 0$ represents a hard spring [64]. However, when $\epsilon < 0$, it represents a soft spring and is given by:

$$\ddot{\zeta}_1 + \gamma\dot{\zeta} + (\kappa + \epsilon\zeta^2)\zeta = 0. \tag{3.10}$$

Converting $x = \zeta$ and $y = \dot{\zeta}$ transforms Eq. (3.10) to

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\gamma x_2 - \kappa x_1 - \epsilon x_1^3. \end{aligned} \tag{3.11}$$

Here, the Duffing oscillator was generated using the parameter values for which the equations do not represent chaotic behaviour: $\kappa = 1$, $\gamma = 0.1$, and $\epsilon = 5$ [64]. Figure 3.26 state variables $x_1(t)$ and $x_2(t)$ were developed using a random uniform distribution containing 100 values between $[-2, 2]$, $[-6, 6]$, respectively. These initial conditions help the oscillator expand to its true form with enough observations ($n = 5000$) and ensure that it is contained within the data for identification.

Figure 3.27 shows that ARGOS consistently represented the Duffing oscillator with high accuracy. However, when the available data was insufficient, the novel approach developed overly sparse models that inadequately captured the dynamics

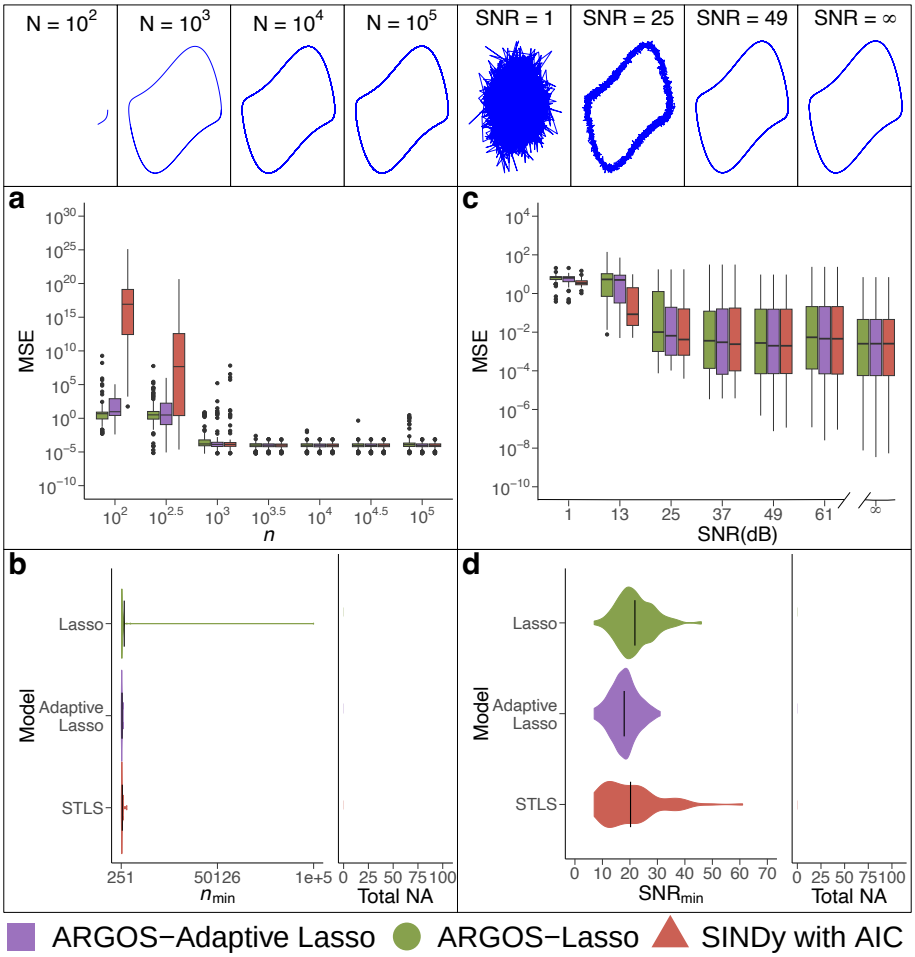

Figure 3.25: **MSE and minimum identification for the Van der Pol oscillator.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

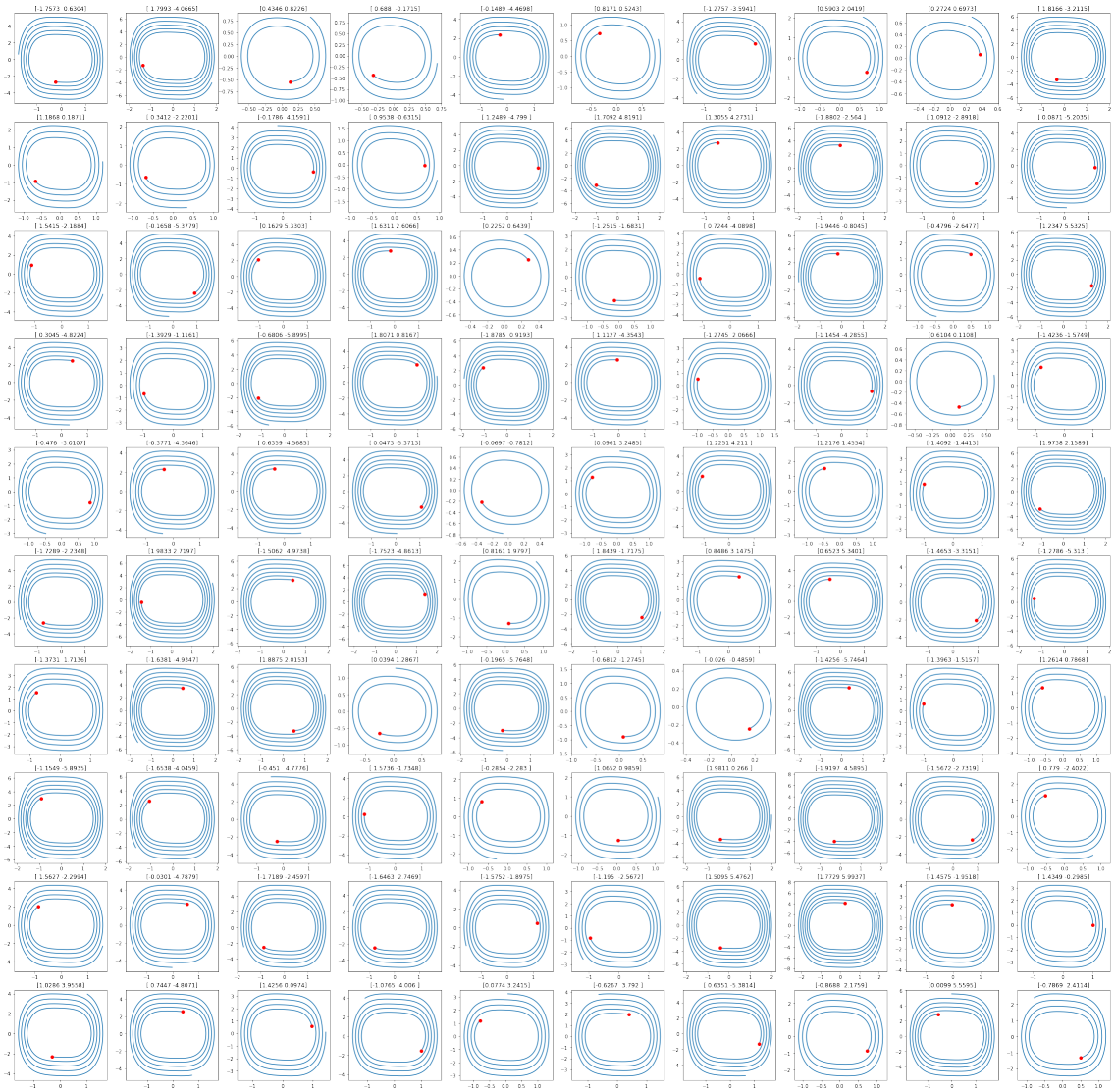

Figure 3.26: **100 Instances of the Duffing oscillator.** The equations were expanded using a random uniform distribution containing 100 values between $[-2, 2]$, $[-6, 6]$ for the state variables $x_1(t)$ and $x_2(t)$, respectively.

of the system. In contrast, SINDy with AIC struggled to discover the system from limited and noisy data, identifying numerous erroneous terms misrepresenting the dynamics.

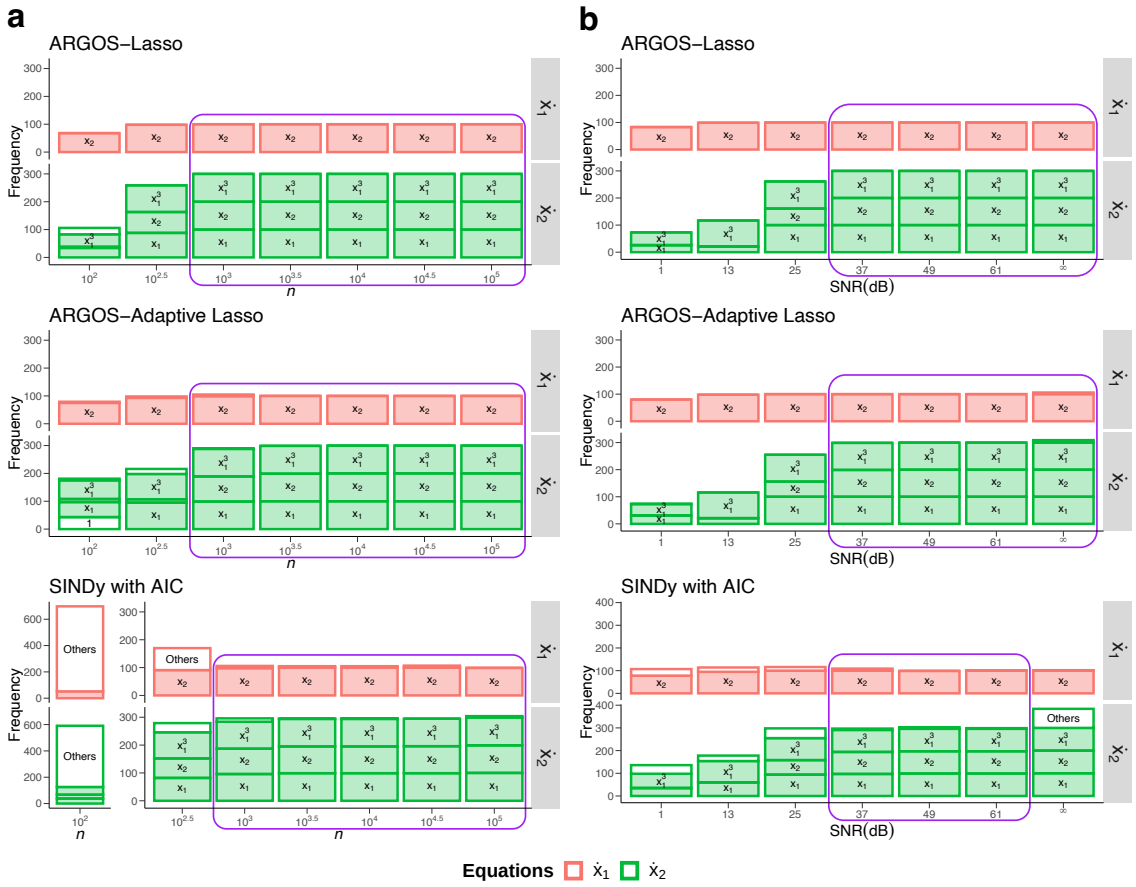

Figure 3.27: **Frequency of identified variables for the Duffing oscillator across algorithms.** Colours correspond to each governing equation; filled boxes indicate correctly identified variables, while white boxes denote erroneous terms. Panels show the frequency of identified variables for data sets with (**a**) increasing n (SNR = 49 dB), and (**b**) SNR (n = 5000). Purple-bordered regions demarcate model discovery above 80%.

Although each algorithm eventually extracted the governing terms of the Duffing oscillator as n increased, ARGOS displayed lower MSE values than SINDy with AIC with fewer observations (Figure 3.28(**a**)). Furthermore, Figure 3.28(**b**) shows that ARGOS required significantly fewer observations to extract the system and continued to do so as the time series length expanded.

Figure 3.28(**c**) exhibits that each algorithm performed similarly well when uncovering prediction models representative of the Duffing oscillator. However, AR-

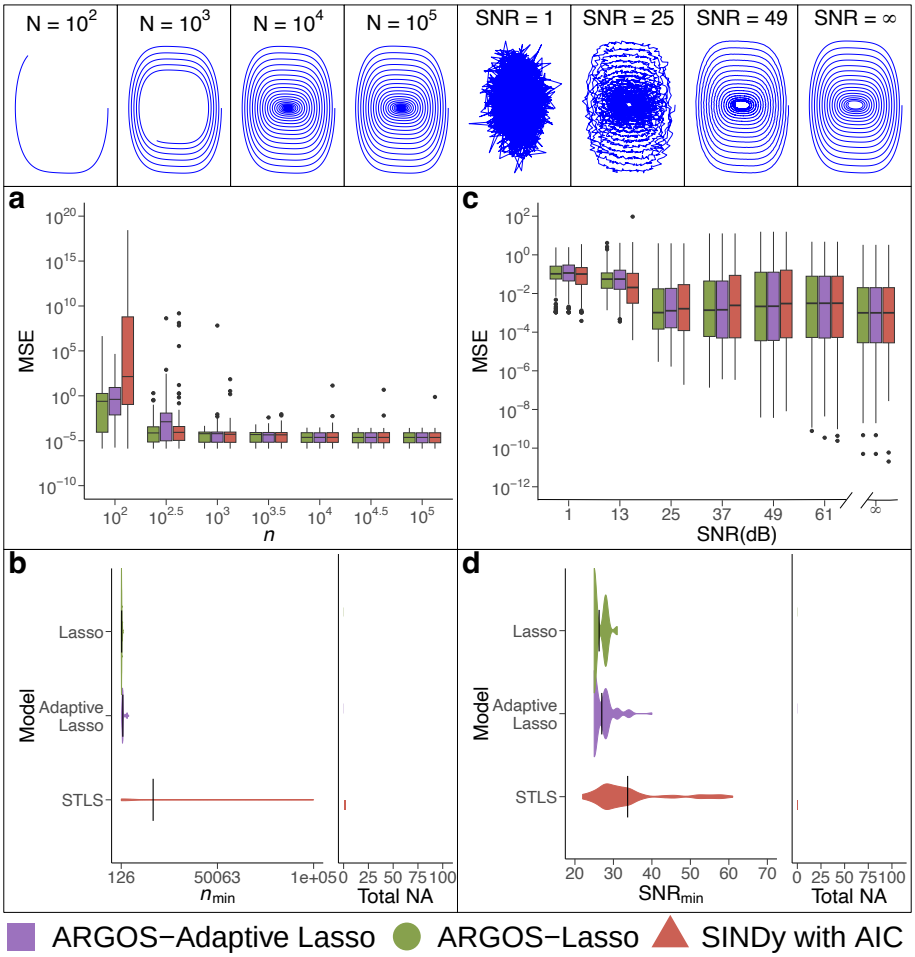

Figure 3.28: **MSE and minimum identification for the Duffing oscillator.** The figure is vertically divided into two regions: keeping SNR = 49 dB constant while increasing n in the left column. (**a**) Provides the distribution of MSE values of each algorithm and (**b**) the average number of observations $\bar{n}_{\min}$ necessary for each algorithm to identify the system as the size of the time series increases. The right column (**b**) shows the distribution of MSE values for each algorithm as the SNR increases in the data. (**e**) shows each algorithm's minimum average $\overline{\text{SNR}}_{\min}$ before it begins to misidentify the system as noise contaminates the system, fixing the number of observations $n = 5000$.

GOS had a lower tail in its distribution at higher values of SNR. Figure 3.28(**d**) further shows that ARGOS was more robust than SINDy with AIC, developing a lower $\overline{\text{SNR}}_{\text{min}}$ value for the system. Therefore, ARGOS provides a new, noise-robust method for automatically identifying these second-order equations from data.

3.4.4 New England Bus System

Forced oscillations are a critical concern for the stability of power systems. Effective identification and mitigation strategies are essential to safeguard against the operational risks they pose [138]. The reliability of power systems depends heavily on the accurate location of the source of these oscillations. Consequently, this section employs ARGOS to trace the origins of such disturbances within the ISO New England system, using data from the IEEE Task Force's test case library documenting actual oscillation events [138].

The behaviour of power systems is complex and often characterised by nonlinear Differential Algebraic Equations. These equations are particularly important when analysing the dynamics of power generators under normal operating conditions. Thus, the swing equations provide a standard approach to modelling generator rotor dynamics, reflecting the balance of power within the system and the influence of external disturbances:

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2, \\ \Psi\dot{\mathbf{x}}_2 &= \mathbf{P}_{mech} - \mathbf{P}_{elec} - D\mathbf{x}_2 + \mathbf{u}, \end{aligned} \tag{3.12}$$

where vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ correspond to the rotor angles and angular velocities of the generators, respectively [138]. The equation parameters are defined as follows: $\Psi = \text{diag}[\Psi_1, \ldots, \Psi_m]$ is the matrix of inertia constants for each generator, providing insight into their respective abilities to resist changes in rotational speed. Similarly, $D = \text{diag}[D_1, \ldots, D_m]$ denotes the damping coefficients essential in offsetting power fluctuations and maintaining operational equilibrium. The vectors $\mathbf{P}_{mech}$ and $\mathbf{P}_{elec}$ represent the generators' mechanical and electrical power outputs. Furthermore, the term $\mathbf{u}$ signifies the external inputs that may include periodic forces exerted on the

generator shafts, such as those arising from maladjustments in control systems like turbine governors or exciters. In the study of forced oscillations, the external input $\mathbf{u}$ is particularly interesting as it may embody the periodic forcing factors that disrupt generator operations [138]. Such detailed examination of the swing equations and their parameters affords deeper insight into potential disturbances, enabling more effective preventive measures to be developed and implemented.

Cai et al. [138] have demonstrated that the forced oscillations, arising from the excitation systems or turbine governors, can manifest themselves in the swing equations given by Eq. (3.12). The periodic inputs, denoted by $\mathbf{u}(t)$, exhibit regular fluctuations that are aptly characterised by the Fourier series. This analytical approach transforms the periodic functions into the sum of sinusoidal components for each generator, expressed as:

$$\begin{aligned} u_j(t) &= \sum_{i=1}^{l} \left(a_{ij} \sin\left(\omega_{F_{ij}} t\right) + b_{ij} \cos\left(\omega_{F_{ij}} t\right) \right) \\ &= \sum_{i=1}^{l} \left(\sqrt{a_{ij}^2 + b_{ij}^2} \sin\left(\omega_{F_{ij}} t + \varphi_{ij}\right) \right. \\ &= \sum_{i=1}^{l} \left(\sqrt{\zeta_{ij}} \sin\left(\omega_{F_{ij}} t + \varphi_{ij}\right) \right), \end{aligned} \tag{3.13}$$

where $\zeta_{ij} = \sqrt{a_{ij}^2 + b_{ij}^2}$ is the resultant amplitude and $\varphi_{ij} = \arctan\left(\frac{b_{ij}}{a_{ij}}\right)$ is the phase angle for each harmonic component within the generator's input.

The shift from deterministic to stochastic modelling is critical in power system analysis, as it allows for including inherent uncertainties in operational dynamics. Such uncertainties often arise from load power variations, modelled as stochastic elements characterised by Gaussian noise. These stochastic variations are a fundamental aspect of the system's behaviour at the generator's internal buses [138]. To account for these load variations, the swing equations are expanded as

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2, \\ \Psi \dot{\mathbf{x}}_2 &= \mathbf{P}_{mech} - \mathbf{P}_{elec} - D\mathbf{x}_2 - V^2 G \Sigma \boldsymbol{\eta} + \mathbf{u}. \end{aligned} \tag{3.14}$$

Here, V represents the voltage magnitude at the generator's internal buses, while

G and Σ denote the conductance matrix and the standard deviations of the load variations, respectively. The vector $\boldsymbol{\eta}$, composed of standard Gaussian random variables, correlates to the noise in load power. This extension of the swing equation aims to encompass the stochastic nature of power systems, providing a more robust and realistic model that better reflects the unpredictable aspects of power system operation.

The structured framework of a physical power system model provides a methodical approach to isolating and assessing the components indicative of forced oscillations, as detailed in the approach by Cai et al. [138]. Here, ARGOS is applied to data gathered from Phasor Measurement Units (PMU), facilitating the extraction of underlying terms that govern oscillatory behaviour. Such an assessment is instrumental in pinpointing the origins of forced oscillations, which is critical in maintaining system stability and reliability.

By integrating the expression for $\mathbf{u}(t)$ from Eq.(3.13) into the linearised stochastic dynamic model of a power system shown in Eq.(3.14), the resulting model, incorporating the periodic input vector, is depicted by the following state-space representation:

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\Psi^{-1}V^2G\Sigma\boldsymbol{\eta} \\ 0 & -\Psi^{-1}\frac{\partial \mathbf{P}_{elec}}{\partial \mathbf{x}_1} \\ I & -\Psi^{-1}D \\ 0 & \mathbf{a}_1 \\ \vdots & \vdots \\ 0 & \mathbf{b}_1 \\ 0 & \mathbf{b}_p \end{bmatrix}^T \begin{bmatrix} 1 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \sin\left(\omega_{F_1}t\right) \\ \cos\left(\omega_{F_1}t\right) \\ \vdots \\ \sin\left(\omega_{F_p}t\right) \\ \cos\left(\omega_{F_p}t\right) \end{bmatrix} \tag{3.15}$$

where ω_{Fi}, $i = 1, \ldots, p$ represents the dominant frequencies of forced oscillations affecting the power system, while a_{ij} and b_{ij} quantify the oscillation input magnitudes at the jth generator for each frequency ω_{Fi} [138].

The next phase involves transforming the model to focus solely on the magnitudes

of these oscillation frequencies, leading to the formation of the coefficient matrix $\mathbf{B}_{ab}^{T}$:

$$\mathbf{B}_{ab}^{T} = \begin{bmatrix} 0 & \cdots & a_{11} & \cdots & a_{1m} \\ \vdots & \cdots & b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \cdots & a_{p1} & \cdots & a_{pm} \\ 0 & \cdots & b_{p1} & \cdots & b_{pm} \end{bmatrix}. \tag{3.16}$$

When the power system is under the influence of forced oscillations, the matrix $\mathbf{B}_{ab}$ exhibits significant non-zero values only at entries corresponding to the source generators of the oscillations.

Cai et al. apply the Fast Fourier Transform (FFT) to the state matrix to identify potential sources of oscillation frequencies [138]. This identification is refined using the z-score-based peak detection method, which isolates the primary candidates for the source of forced frequencies ω_{Fi}. The presence of peaks in the FFT spectrum indicates possible sources of disturbances, which are subsequently confirmed by computing the squared magnitudes ζ_{ij}:

$$\zeta_{ij} = a_{ij}^{2} + b_{ij}^{2} \quad i = 1, \ldots, p, \quad \text{and} \quad j = 1, \ldots, m. \tag{3.17}$$

The calculated ζ_{ij} values, derived from $\mathbf{B}_{ab}$, provide a measure of each oscillation's impact across frequencies and generators. In this setting, high ζ values correspond to the frequency and generator locations of forced oscillations, thereby facilitating the precise localisation of the oscillation sources.

The effectiveness of ARGOS for power system analysis was evaluated using a case study of forced oscillations within ISO-New England's bus network. This analysis encompassed three key generators: G1 at Substation 6, G2 at Substation 7, and G3 at Substation 8 [138]. The study also considered the network's interactions with two external areas, designated as "Area 2" and "Area 3". Here, Lines 7 and 21, located at Substation 3 and Substation 9, respectively, served as the cumulative interface points for these areas. Without rotor angle readings, voltage angle and frequency measurements from the terminal buses of the aforementioned generators,

as well as from the interface buses, were used. Despite this adjustment potentially affecting the Jacobian matrix estimation, the integrity of forced oscillation data within the $\mathbf{B}_{ab}$ matrix remains intact. The system's reference point was set at Line 11 in Substation 5 [138]. Importantly, the framework proposed by Cai et al. [138] keeps the raw PMU data intact, which showcases ARGOS' resilience to measurement noise, temporal data shifts, and various real-world inaccuracies.

Figure 3.29 illustrates the effectiveness of the ARGOS-Adaptive Lasso in identifying the source of forced oscillations across various cases. The method accurately located the source within external Area 2—Line 7 for Cases 1 and 4, as depicted in panels (a) and (c) of the figure. On July 20th, 2017, Case 3 presented as a regional oscillation event, with the system's response modes being thoroughly documented [138]. ARGOS-Adaptive Lasso correctly identified the source proximal to G2, while ARGOS-Lasso recognised the significance of G2's 1.13Hz oscillation frequency. Case 5's oscillation, also linked to G2, was successfully identified, underscoring the method's consistent performance in real-world applications and its automated system identification capabilities.

In summary, the application of ARGOS to the ISO New England system has provided valuable insights into the identification of forced oscillation sources within a complex power network. By employing ARGOS to extract the stochastic swing equations directly from PMU data, the resulting magnitudes elucidate the system's dynamics and susceptibility to operational disturbances. The case studies presented demonstrate ARGOS' robustness in addressing real-world data imperfections and its precision in localising the source of oscillations. Such precise identification is crucial for developing targeted mitigation strategies and reinforcing the grid's operational reliability and stability. Future work will build on these findings, exploring the scalability of the approach to larger systems and its integration with real-time monitoring for proactive disturbance management. The continuous enhancement of such analytical tools remains a cornerstone in the endeavour for power system resilience, affirming the indispensable contribution of ARGOS to this critical field.

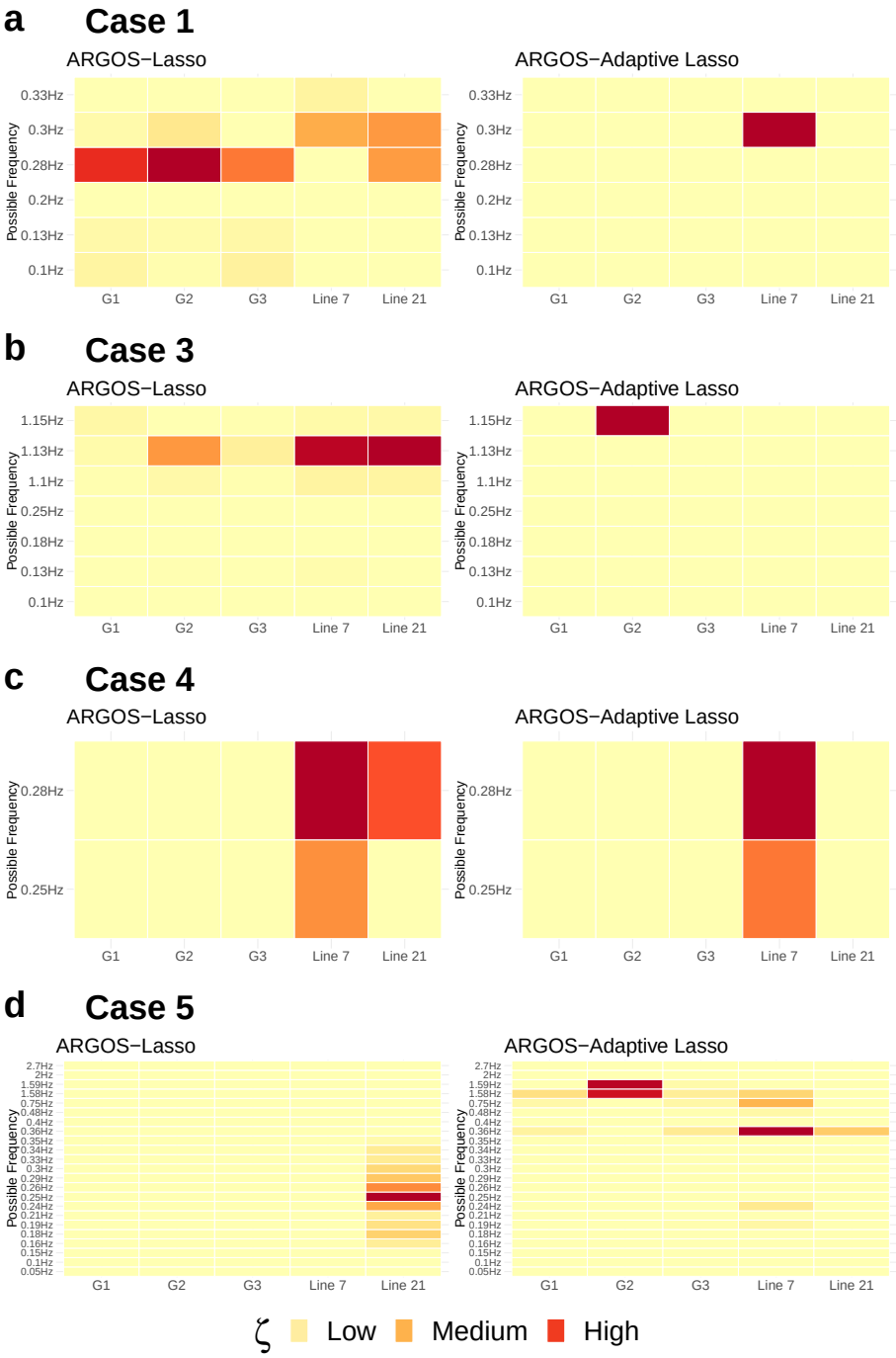

Figure 3.29: **ARGOS Analysis of Forced Oscillation Sources in the ISO-New England Bus Network.** Panels (a) to (d) display ζ parameter heatmaps for Cases 1, 3, 4, and 5, using ARGOS-Lasso and ARGOS-Adaptive Lasso. The data encompasses oscillation sources from generators G1, G2, and G3 and external lines from "Area 2" and "Area 3". Results for Case 2 are not depicted due to missing data from G1 and G3. Heatmaps depict the geographic location and intensity of potential sources of oscillation. A colour spectrum from cool to warm denotes oscillation magnitude, with the efficacy of ARGOS-Adaptive Lasso highlighted by its precise source identification.

3.5 Summary

- ARGOS provides scientists and engineers with a reliable approach to performing system identification. The novel method combines the lasso and the adaptive lasso with bootstrap sampling to develop robust confidence intervals and select the true governing terms from data securely.

- SINDy with AIC is a recently developed approach that performs sequential threshold least squares and determines the optimal model with AIC [67]. Although the method automates model selection, it requires prior knowledge of the governing equations and does not determine the numerical derivative automatically.

- The optimal Savitzky-Golay filtering approach proposed here enables ARGOS and SINDy with AIC to automatically perform the entire system identification process. The method develops a grid of window lengths l and uses polynomial order $o = 4$ before identifying the optimal l^* that minimises the MSE between the original and smoothed signal.

- Through a systematic analysis, this chapter highlights the ability of ARGOS, showing that the method more frequently identifies the underlying equations from data than SINDy with AIC. Furthermore, ARGOS represents a novel process for system identification and encourages researchers to apply statistical inference methods to discover dynamical systems from data automatically.

- Applying ARGOS within the ISO New England system demonstrated its capability to identify sources of forced oscillations, using raw PMU data to address real-world inaccuracies such as measurement noise and temporal data shifts. This application underscores ARGOS' effectiveness in enhancing the reliability and stability of power systems through precise source localisation and effective disturbance mitigation.

CHAPTER 4

Clustering-Based Methods for the Sparse Identification of Nonlinear Dynamics

Although the SINDy algorithm has extended the field, the approach has limitations, primarily lacking a fully automated process for determining the appropriate sparsity-promoting parameter λ_{SINDy} for system identification. Furthermore, the manual selection of these hyperparameters is both time-consuming and susceptible to human error. To overcome this limitation, a more standard method for model evaluation, such as grid-search cross-validation, becomes essential. By systematically assessing several combinations of parameters, this model evaluation approach can identify the optimal parameters, enhancing the accuracy and reliability of the resulting models. This chapter builds on this idea by introducing ASINDy, an adaptive extension that automates the SINDy algorithm. ASINDy uses several clustering-based methods, namely Otsu's method and the K-means algorithm, to identify an initial sparsity-promoting parameter before developing a grid of threshold values for cross-validation, forming the cornerstone of its automated hyperparameter tuning process.

The following chapter demonstrates the effectiveness of ASINDy for several nonlinear systems, showing that it can automatically identify complex equations from

data. With this computationally efficient approach, engineers can study the behaviour of mathematical models more quickly and accurately than with manual processes. Furthermore, by automating the tuning process with grid-search cross-validation, ASINDy standardises the method for model evaluation, reducing the risk of overfitting and optimising model selection.

4.1 Methods

4.1.1 Adaptive-SINDy

Similar to ARGOS, ASINDy begins by applying the optimal Savitzky-Golay filtering algorithm for data smoothing and derivative approximation of each state matrix column $\tilde{\mathbf{x}}_j$ (see Section 3.1.1). The design matrix $\boldsymbol{\Theta}(\mathbf{X})$ construction also mirrors the ARGOS framework, with the addition of trigonometric functions for systems containing Fourier transformations.

Next, ASINDy estimates each column of $\dot{\mathbf{X}}$ in Eq. (2.51) using OLS regression. The normalised OLS coefficients serve to construct the matrix $\hat{\mathbf{B}}^{\text{OLS}}$ as an initial approximation of the system. With this approximation, ASINDy employs Otsu's method or K-means clustering on the OLS coefficients to determine an initial sparsity-promoting λ_0 value. These strategies are essential for constructing a grid of thresholds used for performing subsequent cross-validation and distinguishing significant coefficient estimates for the underlying system.

Otsu's method maximises the between-class variance in the histogram of the normalised coefficients, separating them into two groups of significant and insignificant estimates. ASINDy applies Otsu's method directly to the normalised coefficients, which facilitates an initial sparsity-promoting parameter λ_0 before establishing a range of values for cross-validation, optimising the sequential thresholding process. On the other hand, ASINDy uses the K-means algorithm to assign each parameter estimate to precisely one cluster and establish significant coefficients between two clusters of estimates. Since the K-means algorithm ensures that the clusters are non-overlapping – no observation belongs to more than one cluster – this characteristic allows ASINDy to calculate an initial $\lambda_0 = |C_2 - C_1|$, where C_1 and C_2 are the

corresponding cluster centroids [50].

Upon determining the initial λ_0 (via Otsu's method or K-means clustering), ASINDy fine-tunes the sparsity-promoting parameter. This is accomplished by testing a range of 100 evenly spaced λ values in the interval $[\lambda_0/10, 1.1{\cdot}\lambda_0]$ and identifying the optimal $\lambda^*_{\text{ASINDy}}$ that minimises the 5-fold cross-validation error. ASINDy then uses the optimal $\lambda^*_{\text{ASINDy}}$ as the final sparsity-promoting parameter for its prediction model, enabling the method to solve the problem in Eq. (2.59). By employing grid-search cross-validation, ASINDy improves upon the original SINDy method, determining the optimal model through practical model evaluation rather than a manually selected threshold [46].

Algorithm 3 Adaptive-SINDy

Input: $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times m}$, d.

1: apply Savitzky-Golay Algorithm 1 and build $\boldsymbol{\Theta}(\mathbf{X})$ with functions up to order d;
2: **for** $j = 1$ to m **do**
$\hat{\beta}_j = \text{argmin}_{\beta_j} \|\dot{\mathbf{x}}_j - \boldsymbol{\Theta}(\mathbf{X})\beta_j\|_2^2$;
3: **end for**
4: convert $\hat{\mathbf{B}}^{\text{OLS}}$ to one-dimensional vector $\hat{\beta}^{\text{OLS}}$;
5: obtain λ_0 via K-means or Otsu's method;
6: develop grid $\Lambda = [\lambda_0/10, 1.1 \cdot \lambda_0]$;
7: perform cross-validation to identify $\hat{\mathbf{B}} = \text{argmin}_{\mathbf{B},\Lambda} \left\|\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\mathbf{B}\right\|_2^2 + \Lambda|\mathbf{B}|_0$.

4.2 Results

4.2.1 Data Sets for System Identification

Building on the procedure outlined in Chapter 3, this section employs the same systematic approach to compare the performance of ASINDy with the original SINDy method. Thus, ASINDy builds design matrices $\boldsymbol{\Theta}(\mathbf{X})$ with monomial functions up to $d = 5$ of the smoothed columns of $\mathbf{X}$. However, trigonometric basis functions are also included in the design matrix to account for the Fourier dynamics in the Pendulum motion model and the Thomas system.

The specific implementations of ASINDy are evaluated using Otsu's method and the K-means clustering approach to identify the initial hyperparameter cut-off value

for sequential thresholded least squares and ridge regression methods. ASINDy then applies 5-fold cross-validation to determine the optimal model, assessing a range of regularisation tolerance values $\alpha_{tol} = [0, 0.01, 0.05, 0.1, 0.5]$ when performing sequential thresholded ridge regression. Finally, each method's success rate is calculated and compared against the original SINDy algorithm using its default parameters [46].

4.2.2 Lotka-Volterra System

Two-dimensional nonlinear systems offer a simplified, more tractable model for understanding complex phenomena, allowing engineers to isolate specific properties and behaviours that additional dimensions would otherwise obscure. Thus, the predator-prey equations of the Lotka-Volterra system were re-examined to determine the efficacy of ASINDy in model discovery, using the same random initial conditions as Chapter 3 Section 3.4.2.

Figures 4.1 and 4.2 illustrate the initial challenge of discovering erroneous terms during the sequential thresholding process. However, as the length of the time series and SNR increased, ASINDy eventually attained strong success rates, as shown in Figure 4.5(a). The interaction terms in the design matrix posed added challenges for OLS regression, making it difficult for Otsu's method and the K-means algorithm to distinguish a suitable initial threshold value.

Yet, in spite of these initial challenges, the method developed promising results for model discovery. The figure shows that the K-means method, when used in conjunction with the grid-search cross-validation process employed by ASINDy, consistently discovered these predator-prey dynamics with a high level of accuracy as n and SNR increased. Furthermore, ASINDy performed similarly to the original SINDy method, identifying the system with success rates above 80% even when contaminated with moderate levels of noise in the data. Thus, the proposed approach effectively demonstrated its ability to automate the discovery of this two-dimensional system.

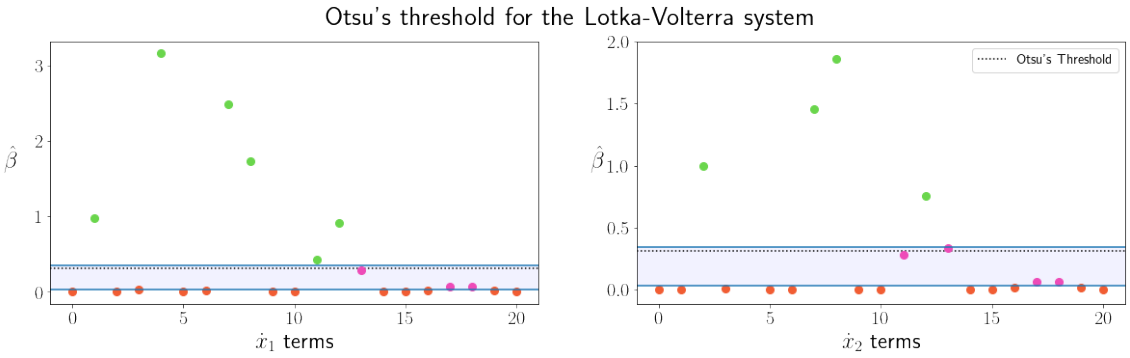

Figure 4.1: **Thresholding estimates for the Lotka-Volterra system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

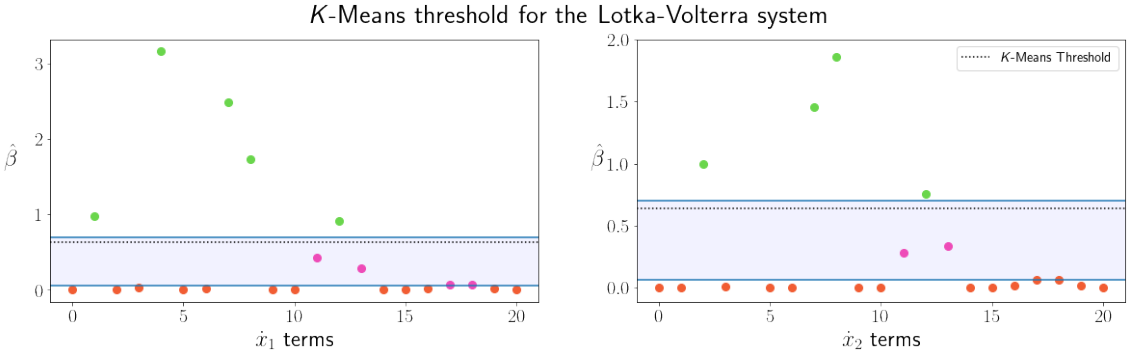

Figure 4.2: **Thresholding estimates for the Lotka-Volterra system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

4.2.3 Chaotic Systems

Chaotic systems present a rich source of complex behaviour to scientists, appearing unpredictable but governed by deterministic laws. The Halvorsen attractor is an example of such a system, featuring chaotic flows that involve cyclic interchanges of the symmetrical state variables. Its governing equations are first-order and exhibit characteristics similar to those found in mechanical, electrical, and biological systems, making the dynamics versatile for studying a range of applications. The equations are represented as

$$\begin{aligned} \dot{x}_1 &= -\alpha x_1 - 4x_2 - 4x_3 - x_2^2, \\ \dot{x}_2 &= -\alpha x_2 - 4x_3 - 4x_1 - x_3^2, \\ \dot{x}_3 &= -\alpha x_3 - 4x_1 - 4x_2 - x_1^2, \end{aligned} \tag{4.1}$$

where $\alpha = 1.27$ [139]. Furthermore, the system was expanded by generating 100 random values for $x_1(t)$, $x_2(t)$, and $x_3(t)$ from a uniform distribution in the range $[-4, 4]$.

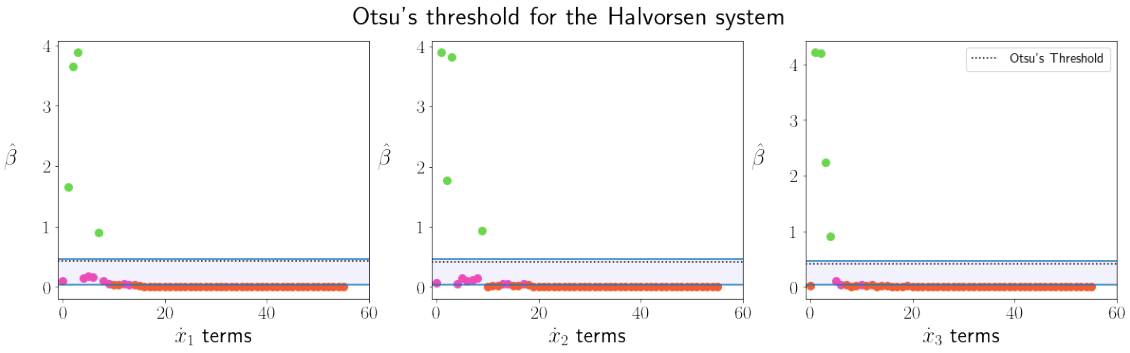

Figure 4.3: **Thresholding estimates for the Halvorsen system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

Both Otsu's and K-means thresholding methods proved to be effective approaches for the proposed technique, as demonstrated in Figures 4.3 and 4.4. Sepa-

rately, these two algorithms successfully determined a value to establish a grid that distinguished between significant and non-significant terms in the regression coefficients, leading to a more accurate representation of the Halvorsen system. Even in the absence of cross-validation, these methods managed to extract the optimal value for the sequential thresholding approach, enabling the discovery of the true governing equations. Moreover, Figure 4.5(b) shows that ASINDy identified the underlying terms under limited and noisy conditions, providing similar results to SINDy in identifying the Halvorsen system.

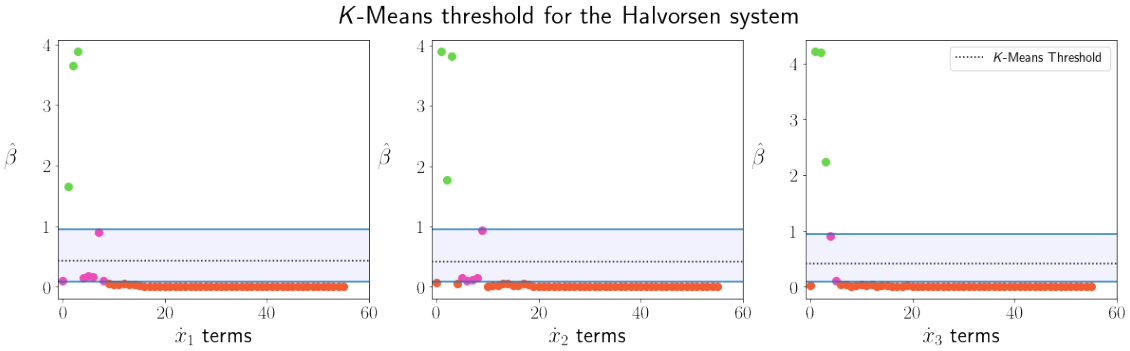

Figure 4.4: **Thresholding estimates for the Halvorsen system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

The next phase of the analysis examined the Lorenz system. As described in Chapter 3, this canonical example of chaos is an excellent candidate for testing the efficacy of any method in model discovery and is used throughout this thesis. Therefore, to examine the system, Eq. (3.8) was expanded using the same parameters and initial condition bounds consistent with Chapter 3.

The thresholding results presented in Figures 4.6 and 4.7 reveal that both Otsu's method and K-means clustering struggled to develop a grid of values that contained optimal threshold values for the sparsity-promoting parameter λ_{ASINDy} in identifying the Lorenz equations, due to the wide range of initial OLS estimates of the system. Nevertheless, the proposed method demonstrated success rates above 80%

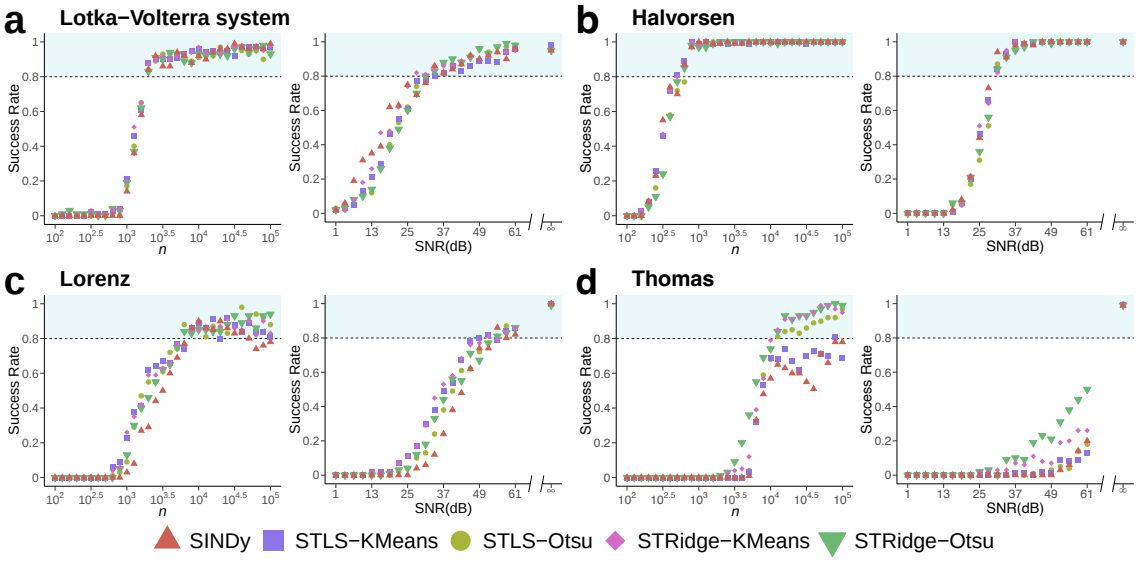

Figure 4.5: **Automatic nonlinear system identification with each proposed ASINDy implementation.** The time-series length n is increased in the system while holding SNR = 49 dB (left panels) and fix $n = 5000$ when increasing the SNR (right panels). Success rates are defined by the proportion of correctly discovered models for each system at each value of n and SNR. First-order nonlinear systems in (**a**) two and (**b**)-(**c**) three dimensions, including (**d**) trigonometric transformations. ASINDy employs K-means clustering and Otsu's method to establish the initial sparsity-promoting tuning parameter for existing sequential thresholded least squares (STLS) [46] and ridge regression (STRR) [47] techniques. Shaded regions represent model discovery above 80%.

as n increased and with high levels of SNR, consistently outperforming the original SINDy algorithm in model discovery (Figure 4.5(c)).

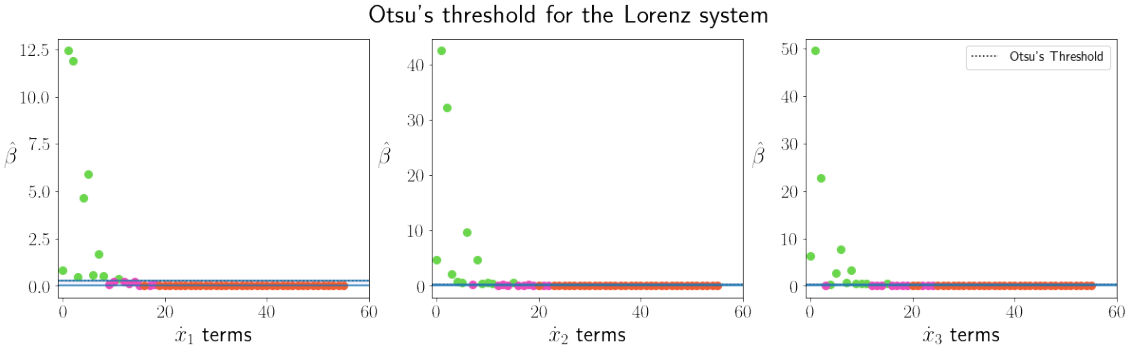

Figure 4.6: **Thresholding estimates for the Lorenz system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

Ultimately, these examples demonstrate ASINDy's effectiveness in automatically representing several examples of chaotic systems from data, offering an advantage over SINDy for discovering three-dimensional dynamics. This advantage is evident in the higher success rates achieved by ASINDy in model discovery, as well as its ability to automate the process of finding the optimal sparsity-promoting parameter, thus potentially saving valuable time for engineers.

4.2.4 Trigonometric Thomas System

Finally, the analysis assessed the identification of a system with Fourier transformations in its nonlinear equations, representing an added challenge for capturing the underlying dynamics. Systems containing trigonometric functions provide a useful mathematical representation of oscillatory and periodic behaviour. In particular, these dynamics are widely used in many areas of science to describe the conduct of waves, signals, and vibrations.

The Thomas system represents a large class of autocatalytic models frequently occurring in chemical reactions. The nonlinear equations governing its dynamics

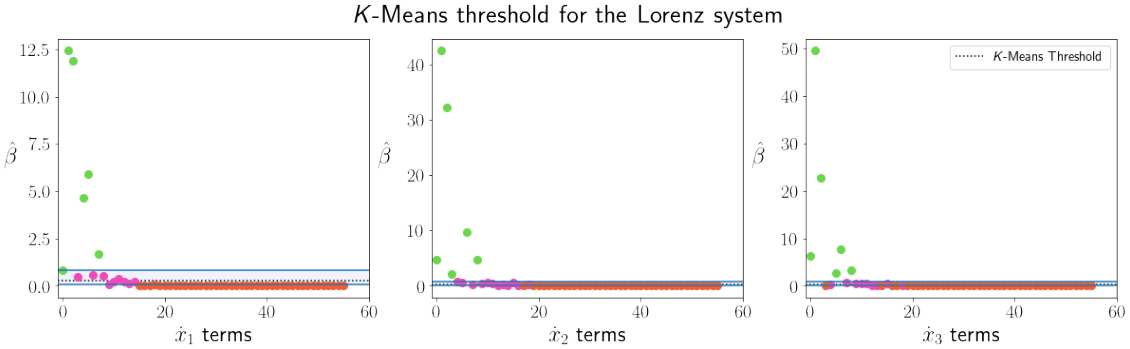

Figure 4.7: **Thresholding estimates for the Lorenz system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

exhibit cyclical symmetry in the state variables and are given by

$$\begin{aligned} \dot{x}_1 &= \sin(x_2) - \zeta x_1, \\ \dot{x}_2 &= \sin(x_3) - \zeta x_2, \\ \dot{x}_3 &= \sin(x_1) - \zeta x_3, \end{aligned} \tag{4.2}$$

where $\zeta = 0.208186$ was used to expand the time series along with a random uniform distribution containing 100 values between $[-1, 1]$ for $x_1(t)$ and $x_2(t)$ [140]. In Eq. (4.2), ζ modulates the Thomas system between two and three dimensions by serving as a damping force for a particle moving in a three-dimensional lattice, influenced by an external energy source or similar resource [140].

Given the trigonometric functions in Eq. (4.2), the design matrix contained candidate trigonometric terms such that

$$\boldsymbol{\Theta}(\mathbf{X}) = \begin{pmatrix} | & | & | & | & | & | \\ \mathbf{1} & \mathbf{X} & \mathbf{X}^{[2]} & \mathbf{X}^{[3]} & \sin(\mathbf{X}) & \cos(\mathbf{X}) \\ | & | & | & | & | & | \end{pmatrix}. \tag{4.3}$$

In Figure 4.5(d), the Thomas system required a sufficiently large time series

length for ASINDy to perform model discovery, while the original SINDy algorithm failed to reach an accuracy above 80%. These results were primarily due to the multicollinearity in the design matrix, which was caused by additional trigonometric functions required to transform the state variables. From these additional terms, Figures 4.8 and 4.9 show that the initial OLS regression estimated higher coefficient values for a significant number of erroneous terms that did not exist in the differential equations. Furthermore, these spurious terms made it difficult for ASINDy to perform sequential thresholding and represent the underlying system successfully, as shown in Figure 4.5(d).

Additionally, when using Otsu's method to establish a range of λ_{ASINDy} values for cross-validation in conjunction with the sequential thresholded least squares method, ASINDy performed best in discovering the underlying equations. Furthermore, sequential thresholded ridge regression clearly reduced multicollinearity, identifying the system more frequently as n increased and with lower values of SNR.

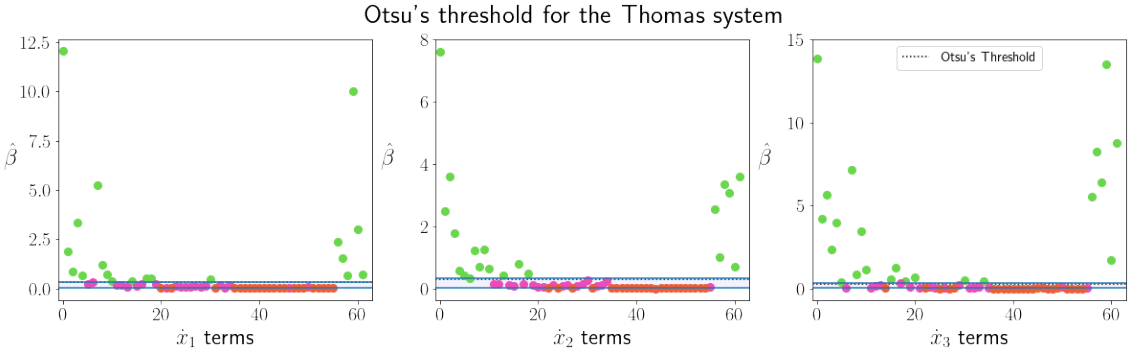

Figure 4.8: **Thresholding estimates for the Thomas system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

The results of each system show the effectiveness of the sequential thresholded ridge regression approach employed by ASINDy, as shown in Figure 4.5. It is important to note that, throughout this study, the original SINDy method used a default sparsity-promoting hyperparameter of 0.1, which often proved to be an efficient ap-

proach for identification. However, the proposed approach demonstrated improved results for the Thomas system, indicating that a larger threshold value was essential for successful model discovery when employing sequential thresholded least squares. Additionally, even though the established method often produced similar results, introducing small levels of regularisation allowed ASINDy to mitigate the effects of multicollinearity slightly, leading to a more consistent system identification by the proposed approach.

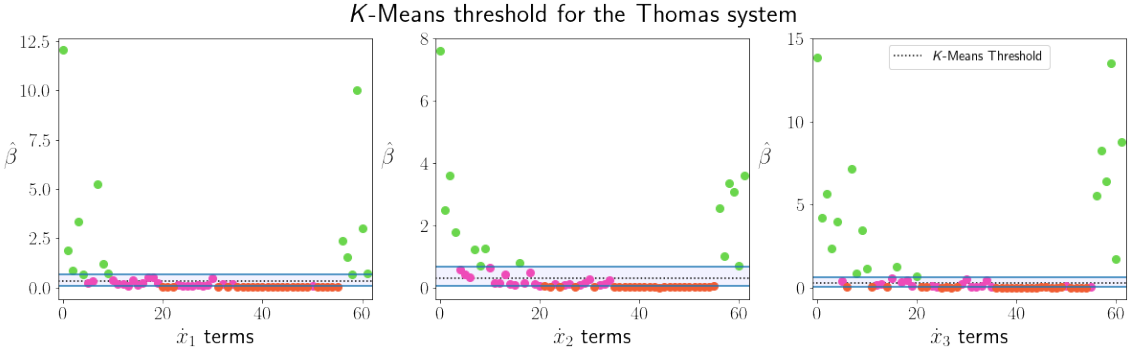

Figure 4.9: **Thresholding estimates for the Thomas system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

4.3 Discussion

The proposed ASINDy approach builds upon the recently developed SINDy algorithm to facilitate its ability to discover nonlinear dynamical systems automatically. ASINDy advances the original sparse regression technique by optimising the parameters for the Savitzky-Golay filter and employing classic unsupervised learning methods to determine an initial sparsity-promoting value. In this process, ASINDy applies the clustering-based thresholding approach of Otsu's method, as well as the K-means algorithm and uses the identified λ value to build a range of thresholds for grid-search cross-validation. These enhancements enable ASINDy to effectively de-

termine an optimal sparsity-promoting parameter for the sequential process, resulting in improved performance for discovering three-dimensional dynamical systems.

Despite these improvements, a critical limitation in all regression-based techniques is the requirement for the active terms to exist in the design matrix when performing identification, and ASINDy is no exception. Additionally, ASINDy may falter when some systems contain very small coefficients, with magnitudes significantly below the threshold, as the identified sparsity-promoting parameter may not be low enough to keep these governing terms in its model. This is where grid-search cross-validation becomes crucial, as it allows for a comprehensive exploration of potential sparsity-promoting parameters, ensuring that ASINDy performs model validation correctly and increasing the likelihood of preserving significant terms even with small coefficients or large initial OLS estimates.

When OLS estimates reasonably approximate the system well, Otsu's method and the K-means clustering approach enable ASINDy to ensure that the true model exists within the subset of developed models. However, when large initial OLS estimates are present, the system may be difficult to discern, relying heavily on the sequential thresholding process of the algorithm. In both settings, engineers must know the systems they aim to discern and understand the potential terms that govern their dynamics. With sufficient data, ASINDy provides a unique approach to determining the optimal sparsity-promoting tuning parameter for many systems, furthering automation in the field, and effectively building more interpretable models from data.

Turning to a comparative evaluation, the systematic analysis of the results highlights the efficacy of ASINDy in identifying various forms of ordinary differential equations, consistently outperforming SINDy in discovering several three-dimensional systems. For instance, the method performs well when the design matrix incorporates Fourier transformations, as demonstrated by its ability to discern the dynamics of trigonometric systems from data. By employing the optimal Savitzky-Golay filter parameters, ASINDy conducts model selection on the smoothed data without making prior assumptions about the underlying structure of a given dynamical system. The approach employs K-fold cross-validation as a standard method for

model evaluation, offering a dependable means of assessing the accuracy of identified models. In contrast, the original approach depends on the identified model without any validation [46]. These results provide a viable alternative to the established technique, particularly in improving the discovery of three-dimensional systems.

While SINDy has significantly impacted the field of system identification, engineers are still required to determine manual tuning parameters to perform model discovery effectively. Here, the adaptive approach employed by ASINDy aims to encourage engineers to seek alternative machine learning methods that help further automate and optimise the conventional system identification framework, enabling them to focus on their data's governing equations rather than spending valuable time determining them manually.

4.4 Additional Case Studies

4.4.1 Van der Pol Oscillator

The Van der Pol oscillator was used to evaluate the efficacy of ASINDy in discovering second-order two-dimensional nonlinear equations from data. For the expansion of the canonical oscillator, $\mu = 1.2$ was again applied to Eq. (3.9), along with the same random uniform distribution containing 100 values between $[-4, 4]$ for $x_1(t)$ and $x_2(t)$.

Applying Otsu's method and K-means clustering yielded efficient results in developing a threshold grid for cross-validation with ASINDy. From the initial estimates, as illustrated in Figures 4.10 and 4.11, it is clear that these two methods determined threshold parameters that successfully guided ASINDy in discovering the Van der Pol oscillator. Additionally, Figure 4.16(a) shows the accuracy of ASINDy in identifying the oscillator, achieving high success rates when using K-means to cluster the initial parameter estimates and developing a grid for determining the optimal threshold.

Interestingly, although Otsu's method and the K-means clustering approach provided noise-robust results for the Van der Pol oscillator, the identification performance using Otsu's method to determine the sparsity-promoting parameter grid

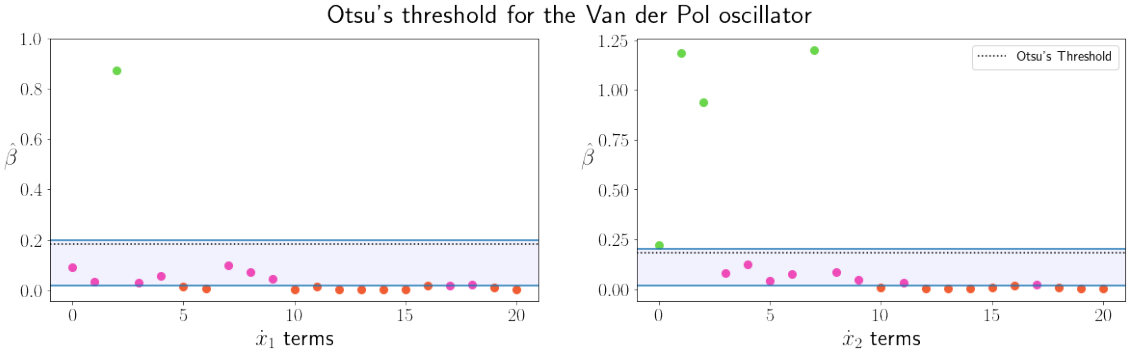

Figure 4.10: **Thresholding estimates for the Van der Pol oscillator using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

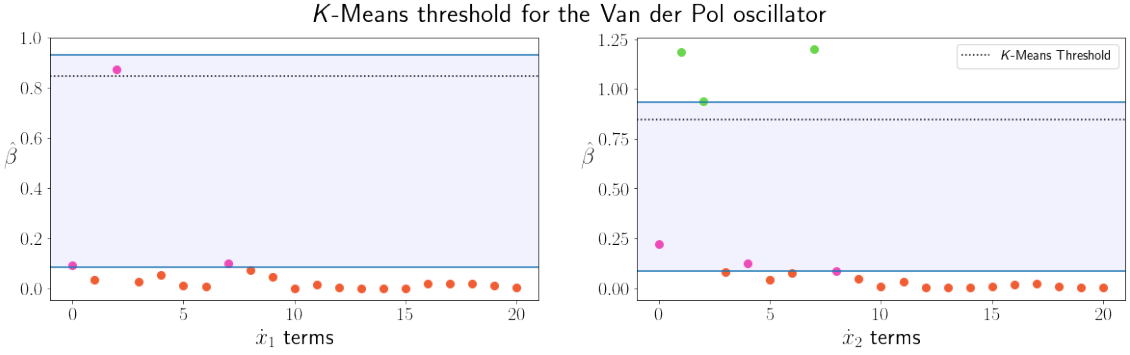

Figure 4.11: **Thresholding estimates for the Van der Pol oscillator using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

for ASINDy with sequential thresholded ridge regression showed a similar drop when the system was noiseless (SNR = ∞). This drop can be attributed to the autocorrelation in the final model's residuals. Under these conditions, the identified model corresponding to the minimum cross-validation error contained several erroneous terms to ensure the residuals were homoscedastic and contained minimal autocorrelation. Ultimately, using grid-search cross-validation to determine the optimal sparsity-promoting parameter was critical for the success of ASINDy in identifying the Van der Pol oscillator.

4.4.2 Dadras system

The Dadras equations, proposed by Dadras and Momeni [141], represent a novel three-dimensional autonomous chaotic system like the Lorenz. The Dadras system can generate two, three, and four-scroll chaotic attractors with a single parameter variation, showcasing its rich nonlinear dynamics, including chaos and period double bifurcations. The governing equations are described as

$$\begin{aligned} \dot{x}_1 &= x_2 - \alpha x_1 + \zeta x_2 x_3, \\ \dot{x}_2 &= \upsilon x_2 - x_1 x_3 + x_3, \\ \dot{x}_3 &= \delta x_1 x_2 - \eta x_3, \end{aligned} \tag{4.4}$$

with the values of the original parameters $\alpha = 3$, $\zeta = 2.7$, $\upsilon = 4.7$, $\delta = 2$, and $\eta = 9$ [141]. Moreover, the state variables were expanded with random uniform distribution containing 100 values between $[-4, 4]$ for $x_1(t)$, $x_2(t)$, and $x_3(t)$.

As a hard-thresholding algorithm, determining the optimal threshold is crucial to the success of ASINDy in automating model discovery. Traditionally, engineers spend time reviewing different models to discover the true form of the underlying system. Figures 4.12 and 4.13 show the effectiveness of thresholding OLS coefficients for this purpose, particularly using Otsu's method, which clearly identified the active terms in the Dadras system when applied to the normalised coefficients. By applying Otsu's method to determine the sparsity-promoting grid, ASINDy identifies the active terms in the Dadras system more consistently than SINDy, as shown

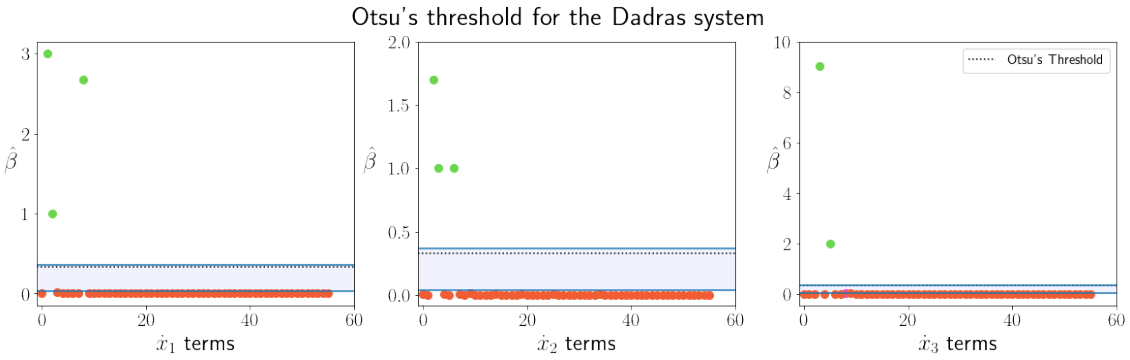

Figure 4.12: **Thresholding estimates for the Dadras system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

in Figure 4.16(b). This allows ASINDy to reliably perform sparse regression with the underlying model in its subset, ensuring that the true model can be identified, given it contains the minimum cross-validation error. Moreover, ASINDy provided a noise-robust method for identifying the underlying system as the SNR increased in the data. Therefore, performing clustering on the initial OLS estimates provides a robust approach for automatically extracting the chaotic Dadras system from data.

The original SINDy algorithm had difficulty identifying the system as the number of observations and SNR increased. As observed with previous examples of chaotic systems, SINDy faces challenges when the assumptions of linear regression are violated. In the case of the Dadras equations, SINDy struggled to derive the underlying equations due to multicollinearity in the design matrix and outliers in the data as the time series expanded. Consequently, the method inadvertently adds erroneous terms to improve its prediction model. Evaluating the integrated state matrix makes the process even more challenging to identify the system, rather than using the traditional approach of comparing the error between the predicted $\dot{x}_j$ equations. Therefore, using K-fold cross-validation enables ASINDy to develop a more straightforward process for discovering equations.

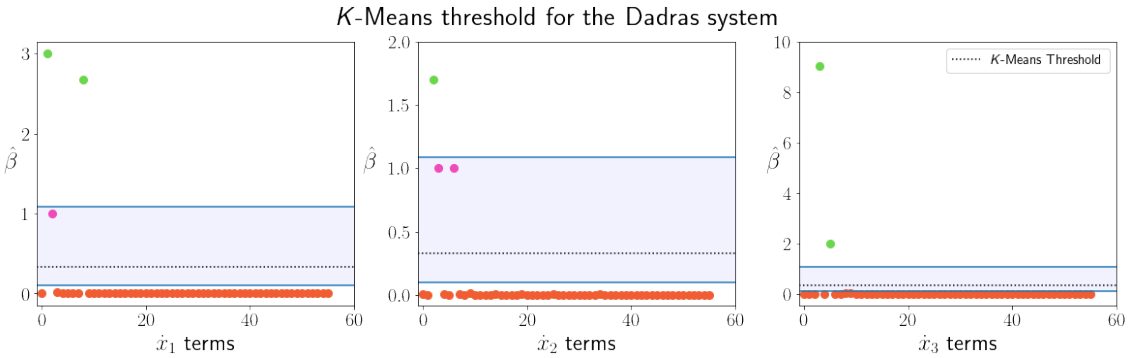

Figure 4.13: **Thresholding estimates for the Dadras system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

4.4.3 Sprott system

The Sprott system, a chaotic attractor that exhibits complex and unpredictable behaviour, is known for its nested collection of invariant tori and quasi-periodic orbits [142]. Furthermore, this system can provide valuable insights for electrical and mechanical engineers due to the wealth of chaotic dynamics it exhibits. The following equations describe the Sprott system:

$$\begin{aligned} \dot{x}_1 &= x_2 + 2x_1x_2 + x_1x_3, \\ \dot{x}_2 &= 1 - 2x_1^2 + x_2x_3, \\ \dot{x}_3 &= x_1 - x_1^2 - x_2^2. \end{aligned} \tag{4.5}$$

Here, the state variables were expanded with a random uniform distribution containing 100 values between $[-1, 1]$ for $x_1(t)$, $x_2(t)$, and $x_3(t)$.

Thresholding methods such as Otsu's method and the K-means algorithm can significantly contribute to the discovery of the Sprott system, as evidenced in Figures 4.14 and 4.15. These methods allowed for a sparse representation without necessitating the entire sequential thresholding process, indicating the potential for effective system identification using ASINDy.

Additionally, Figure 4.16(c) illustrates ASINDy consistently outperforming the established SINDy as the number of observations n and SNR increased. Interestingly, the sequential thresholded ridge regression implementation within ASINDy provides the most accurate representation of the system for both tests, suggesting that slight regularisation facilitates the success of the algorithm.

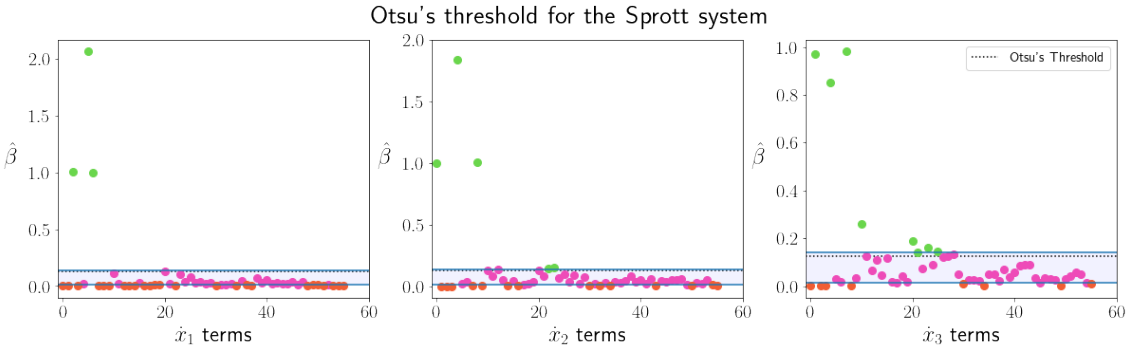

Figure 4.14: **Thresholding estimates for the Sprott system using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

In summary, the analysis of the Sprott system shows that thresholding methods, particularly Otsu and K-means, are effective in revealing the underlying dynamics. Furthermore, the results highlight the critical role of grid search cross-validation in automating the model discovery process and emphasise ASINDy as a more noise-robust and effective method for model evaluation.

4.4.4 Nonlinear Pendulum Motion Model

In contrast to the simple linear pendulum model, the nonlinear model provides a more complex system, considering larger angles of oscillation and nonlinearities such as air resistance and friction [101]. The pendulum motion is described by the angle

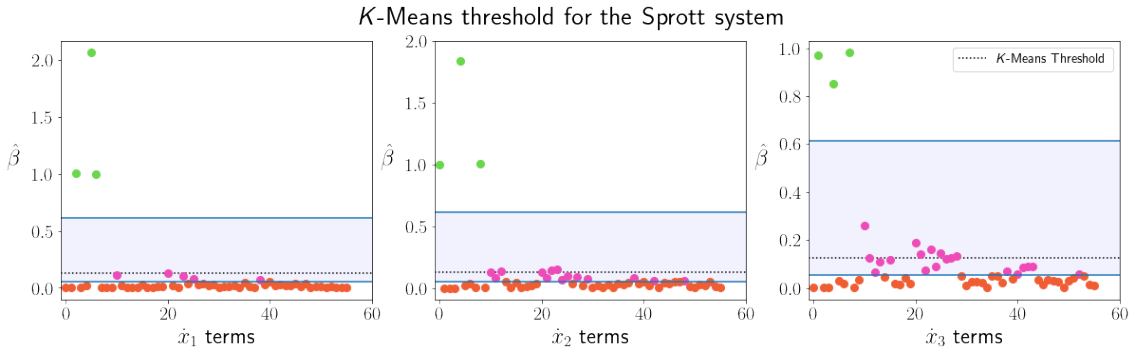

Figure 4.15: **Thresholding estimates for the Sprott system using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

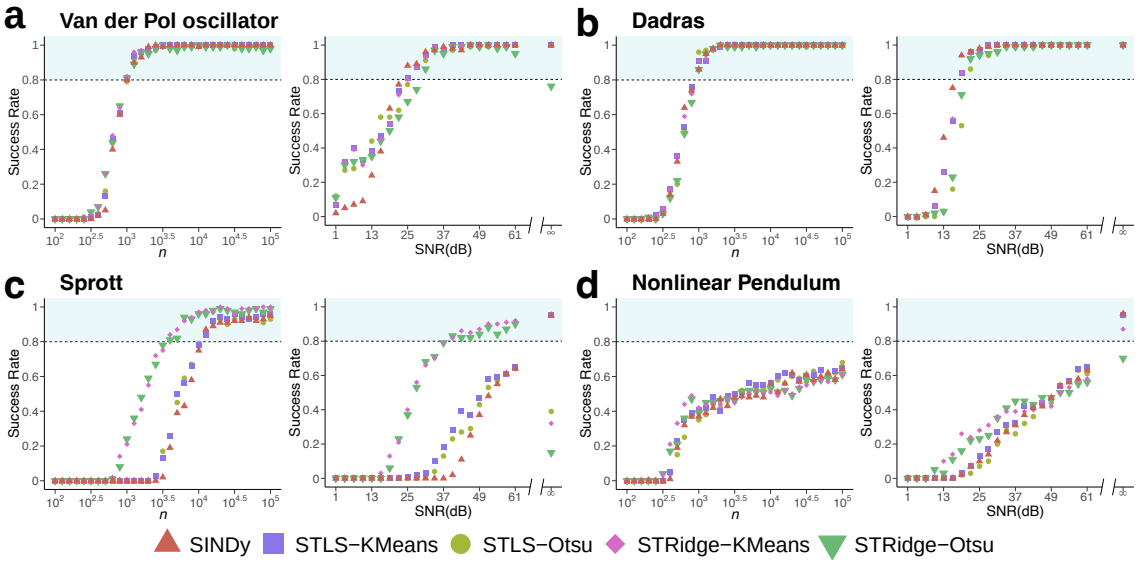

Figure 4.16: **ASINDy identification for nonlinear differential equations.** The time-series length n is increased in the system while holding SNR = 49 dB (left panels) and fix $n = 5000$ when increasing the SNR (right panels). Success rates are defined by the proportion of correctly discovered models for each system at each value of n and SNR. First-order nonlinear systems including (**a**) two and (**b**)-(**c**) three dimensions, as well as (**d**) trigonometric transformations. ASINDy employs K-means clustering and Otsu's method to establish the initial sparsity-promoting tuning parameter for existing sequential thresholded least squares (STLS) [46] and ridge regression (STRR) [47] techniques. Shaded regions represent model discovery above 80%.

$x_1(t)$ and the angular velocity $x_2(t)$ over time t [136]:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\alpha x_2 + \zeta \sin(x_1), \end{aligned} \tag{4.6}$$

where $\alpha = -0.25$ and $\zeta = -5$ [136]. The system was simulated by initializing the state variables using a random uniform distribution. Here, 100 values were selected within the range $x_1(t) = [-1 - \pi, 1 + \pi]$ and $x_2(t) = [-1, 1]$.

Analysis of the performance of ASINDy in identifying the nonlinear pendulum motion model indicates that initial OLS regression estimates resulted in large parameter values for the system. As illustrated in Figures 4.17and4.18, several OLS coefficients are initially above the originally identified threshold value λ_0, leading ASINDy to face challenges reducing these erroneous coefficients to zero during the sequential thresholding process. Consequently, the high coefficients persisted throughout the model identification process.

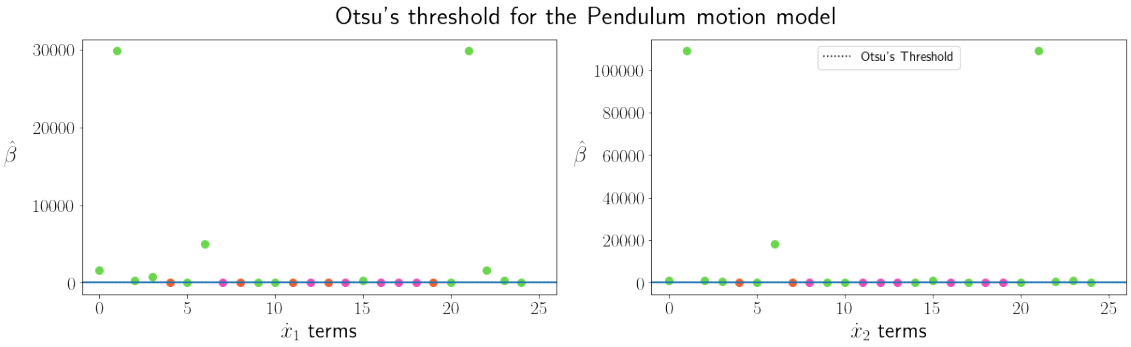

Figure 4.17: **Thresholding estimates for the pendulum motion model using Otsu's method.** Otsu's method is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

Another factor that affects the performance of ASINDy in identifying the pendulum motion model is multicollinearity. Figure 4.16(d) further illustrates the challenges the proposed approach encounters in the presence of system noise. Despite increases in n and SNR, each method struggled to reach an 80% success rate, with

the notable exception of when SNR $= \infty$. The figure illustrates how even small noise magnitudes had a noticeable impact on identification, as ASINDy was able to extract the system with 100% success when the data was noiseless.

Like the Thomas system, engineers should be aware of the impact of multicollinearity on the sequential thresholding methods' ability to discover models containing trigonometric transformations effectively. When contaminated with additional multicollinearity, even when using ridge regression with ASINDy, the active terms may be misrepresented due to initial unstable coefficient estimates, leading the method to misidentify the governing equations. Thus, engineers should be aware of this challenge when identifying systems with Fourier transformations in their governing equations.

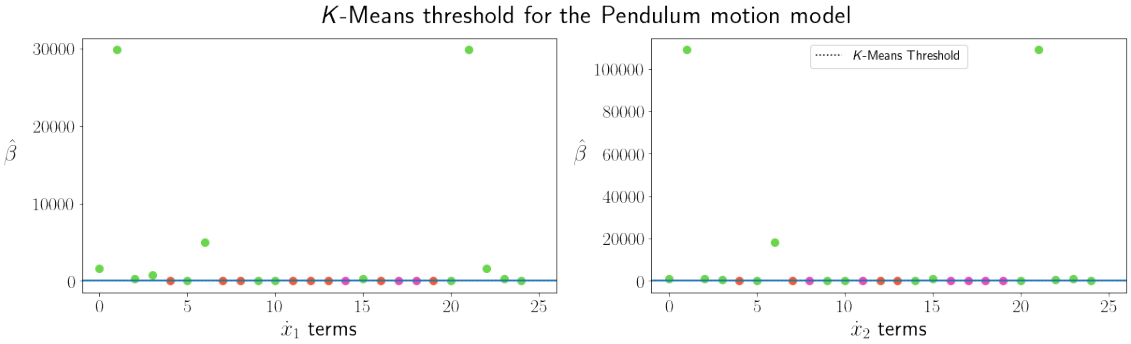

Figure 4.18: **Thresholding estimates for the pendulum motion model using K-means clustering.** K-means clustering procedure is applied to the initial normalised OLS coefficients to establish a grid of threshold values (shown in blue). Coefficients above this grid are marked with green dots, indicating their preliminary selection before ASINDy begins its sequential thresholding process. Purple dots within the grid highlight coefficients potentially relevant for identification. Orange dots outside the grid represent coefficients immediately excluded by all thresholds.

4.5 Computational Time for ASINDy

Figure 4.19 illustrates the computing time required to perform ASINDy when employing Otsu's method for the sparsity-promoting parameter grid in the sequential thresholded least squares and ridge regression processes compared to the original SINDy algorithm [46]. Each algorithm was executed using one CPU core with a

single thread, allowing for a fair comparison of efficiencies in the model discovery process. The figure highlights that the original SINDy approach exhibits linear time complexity, while ASINDy demonstrates quadratic time complexity. This difference can be observed in the steeper growth of computing time for ASINDy as the length of the time series increases. However, despite ASINDy's higher computational demand, its more elaborate approach is beneficial. Integrating the `for` statement in Algorithm 3, followed by k–fold cross-validation, is pivotal in model assessment. This rigorous process ensures that ASINDy evaluates each prediction model's fit to the data, a more comprehensive method than the original SINDy approach. Despite the higher time complexity, the proposed approach consistently outperformed the original SINDy algorithm in identifying complex three-dimensional dynamics, such as the Lorenz system, as evidenced by the success rates previously discussed.

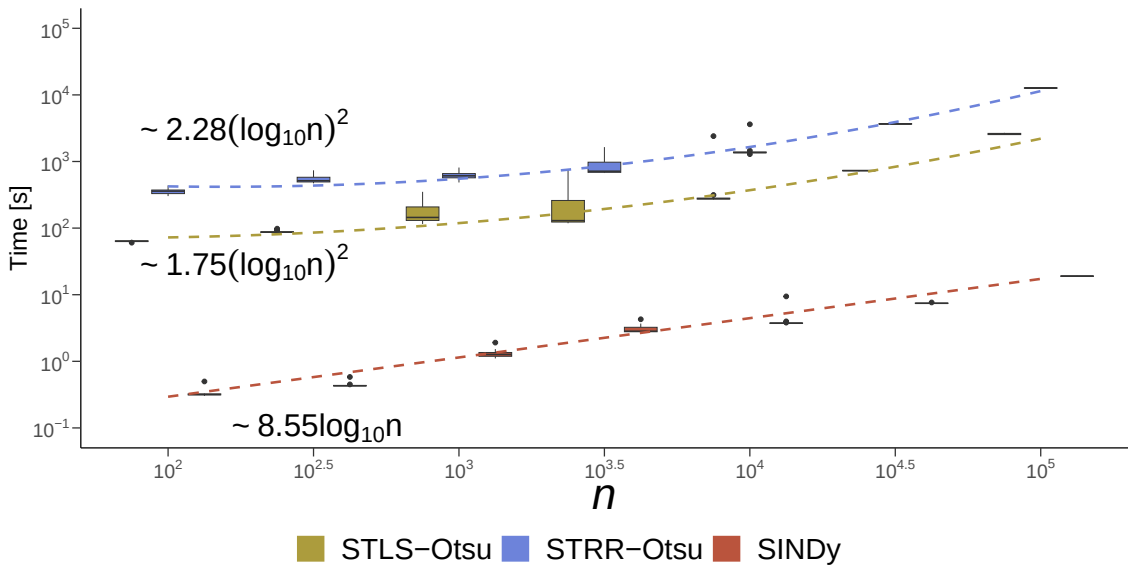

Figure 4.19: **Time-complexity (seconds) between ASINDy and SINDy**. Boxplots depict the computational time required for model discovery over 100 instances for the Lorenz system. The black bar within each box represents the median computational time. Whiskers extending from each box show 1.5 times the interquartile range. Data points beyond the end of the whiskers are outlying points. Equations accompanying the dashed lines indicate the fitted mean computational time for each algorithm at various values of n.

Ultimately, the optimal Savitzky-Golay filtering algorithm can be described as $\mathcal{O}(nw)$, where w represents the grid of window lengths. Furthermore, the sequential

thresholding algorithm within ASINDy displays a similar computational complexity to the lasso, $\mathcal{O}(p^3 + np^2)$. However, the incremental time requirement in ASINDy is justified due to its comprehensive nature, particularly as it employs 5-fold cross-validation across several parameters to determine its optimal model, thus offering a more robust and reliable framework for model evaluation and selection.

4.6 Summary

- Clustering algorithms, such as Otsu's method and K-means, enable ASINDy to group initial OLS parameter estimates together and automatically determine an initial sparsity-promoting grid. With this process, the proposed approach employs the standard cross-validation method for model evaluation to effectively determine coefficient estimates.

- The methods employed by ASINDy provide a computationally efficient and adaptive process that enables researchers to extract the most significant terms from data, often outperforming the original SINDy. This is particularly important for large datasets or systems with high-dimensional state spaces, where alternative techniques may be computationally prohibitive.

- ASINDy implements the optimal Savitzky-Golay filtering method to compute numerical derivatives automatically. Consequently, the signal is smoothed optimally, and the underlying trends become more clearly represented, allowing ASINDy to identify the best-fit prediction models.

- By employing clustering methods for preprocessing, the proposed approach determines a sparsity-promoting parameter grid suitable for sequential thresholding. Furthermore, using the standard K-fold cross-validation technique enables ASINDy to determine the optimal model effectively, as opposed to the original SINDy algorithm, which identifies a single model through an iterative process that lacks a validation set. Thus, this hybrid learning approach enables users to perform system identification reliably and automatically, advancing the original method by carefully evaluating initial estimates and the final model.

CHAPTER 5

A Bayesian Approach to Nonlinear System Identification

This chapter introduces an alternative method, ARGOS-BI, which expands upon the original ARGOS framework by replacing the frequentist bootstrap sampling technique with Bayesian regression for model inference. By employing Bayesian inference, ARGOS-BI offers an efficient and noise-robust process for uncovering dynamical systems from data. The efficacy of ARGOS-BI will be demonstrated for several dynamical systems, showcasing advancements in the ability to automate model discovery of three-dimensional systems, including those involving Fourier transformations in their underlying equations.

In this context, a comparison with the recent Ensemble-SINDy proves advantageous [132]. Chosen for its methodological similarities, Ensemble-SINDy employs bootstrap aggregation (bagging) and posterior probabilities in model identification, akin to the Bayesian underpinnings of ARGOS-BI. This methodological resemblance provides a more precise and robust basis for comparison, highlighting the enhancements and strengths ARGOS-BI introduces to automated model discovery, particularly in terms of robustness and reliability.

5.1 Automatic regression for governing equations with Bayesian Inference

Building upon the original framework in Chapter 3, ARGOS-BI also applies the Savitzky-Golay filtering method to smooth and differentiate the data before developing an initial design matrix $\boldsymbol{\Theta}^{(0)}(\mathbf{X})$ by expanding the columns of $\mathbf{X}$ with monomials up to the d-th degree and any additional basis functions (defined as degree $d = 1$).

The regression procedure in ARGOS-BI includes two versions of the adaptive lasso, one establishing a weights vector w from ridge regression and the other from OLS regression, to develop two potential prediction models concurrently. These models are then refined by reducing their design matrices to include only terms up to the highest order nonzero monomial d of the corresponding estimate. This process is repeated with the updated design matrices, applying OLS to the selected variables of the final models. The model exhibiting the minimum BIC value is then chosen as the optimal model, advancing to the next stage of the algorithm.

In the next stage, ARGOS-BI employs Bayesian regression on the selected variables, enhancing the final model's reliability by addressing inherent uncertainties. This process employs MCMC sampling, which relies on defining a probability function and using independent Gaussian priors to explore the posterior distribution effectively. The preference for Gaussian priors is twofold. Firstly, their mathematical simplicity and computational efficiency in shaping posterior estimations make them particularly useful for ARGOS-BI. In practical terms, these priors facilitate more straightforward calculations and precise interpretations of the model's outputs. This clarity is crucial in ARGOS-BI, where understanding the impact of each variable on the predictions helps in refining the final model and ensuring its accuracy and reliability. Secondly, the choice of Gaussian priors aligns with the insights provided by the central limit theorem. This theorem suggests that, regardless of an initial distribution's shape, the distribution of sample means tends to a normal (Gaussian) distribution as the number of samples increases [93, p.51]. For ARGOS-BI, this implies that employing Gaussian priors is not just mathematically convenient but also

statistically robust, especially when dealing with large data sets where the theorem's effects become pronounced.

After Bayesian regression, the Gaussian prior's effectiveness particularly shines when dealing with data that inherently approximate Gaussian distributions in their noise characteristics or when the central limit theorem's applicability is a reasonable assumption, ensuring not just mathematical convenience but also an alignment with the empirical data's nature. This alignment not only simplifies computations but also enhances the model's predictive accuracy by closely mirroring the underlying statistical properties of the data.

This robustness underpins the reliability of Bayesian regression within the ARGOS-BI framework, providing confidence in the model's performance across varying scenarios and data characteristics. Consequently, the approach aims for the samples to converge to an accurate posterior distribution, capturing the inherent uncertainties in the final model. Significantly, MCMC sampling from this distribution quantifies uncertainty and calculates the posterior medians for each coefficient, which act as precise point estimates in the final model, illuminating the influence of each variable on prediction.

Lastly, ARGOS-BI then constructs credible intervals using the posterior probabilities derived from the Bayesian regression process. Rather than the traditional 95% intervals, ARGOS-BI constructs 90% credible intervals to offer greater stability and reliability in the estimates [97]. Like the ARGOS framework in Chapter 3, the proposed ARGOS-BI finalises the model by selecting variables whose credible intervals do not include zero and whose point estimates fall within these intervals. The detailed step-by-step procedure of this regression process is outlined in Algorithm 4, which encapsulates the complete computational workflow of ARGOS-BI.

5.2 Results

5.2.1 Building the data sets

This study applies the methodology established in Chapters 3 and 4 for creating data sets, aiming to evaluate the effectiveness of ARGOS-BI in performing model

Algorithm 4 Automatic Regression for Governing Equations - Bootstrap Inference (ARGOS-BI)

Input: $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\dot{\mathbf{x}}_j \in \mathbb{R}^n$, d, $\alpha = 0.05$.

▷ STEP ONE: Initial design matrix

$p^{(0)} = \binom{m+d}{d}$;

create $\boldsymbol{\Theta}^{(0)}_{\text{ridge}}(\mathbf{X})$ and $\boldsymbol{\Theta}^{(0)}_{\text{OLS}}(\mathbf{X}) \in \mathbb{R}^{n \times p^{(0)}}$ with basis functions up to order d of the columns of $\mathbf{X}$;

▷ STEP TWO: Determine optimal adaptive lasso model

▷ λ^*_{ridge}, λ^*_{OLS} : Optimal λ from 10-fold cross-validation with ridge and OLS regression weights

▷ $w_{\text{ridge}}, w_{\text{OLS}}$: ridge and OLS regression coefficients

$$\hat{\beta}^{(0)}_{\text{ridge}} = \underset{\beta}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(0)}_{\text{ridge}}(\mathbf{X})\beta \right\|_2^2 + \lambda^*_{\text{ridge}} \sum_{k=1}^{p^{(0)}} w_{\text{ridge},k}|\beta_k|,$$

$$\hat{\beta}^{(0)}_{\text{OLS}} = \underset{\beta}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(0)}_{\text{OLS}}(\mathbf{X})\beta \right\|_2^2 + \lambda^*_{\text{OLS}} \sum_{k=1}^{p^{(0)}} w_{\text{OLS},k}|\beta_k|$$

extract $\boldsymbol{\Theta}^{(1)}_{\text{ridge}}(\mathbf{X})$ and $\boldsymbol{\Theta}^{(1)}_{\text{OLS}}(\mathbf{X})$ to contain columns of $\boldsymbol{\Theta}^{(0)}_{\text{ridge}}(\mathbf{X})$ and $\boldsymbol{\Theta}^{(0)}_{\text{OLS}}(\mathbf{X})$ up to the largest order of the selected variables in $\hat{\beta}^{(0)}_{\text{OLS}}$ and $\hat{\beta}^{(0)}_{\text{ridge}}$;

$$p^{(1)}_{\text{ridge}} = \binom{m + d^{(1)}_{\text{ridge}}}{d^{(1)}_{\text{ridge}}}, \quad p^{(1)}_{\text{OLS}} = \binom{m + d^{(1)}_{\text{OLS}}}{d^{(1)}_{\text{OLS}}}$$

$$\hat{\beta}^{(1)}_{\text{ridge}} = \underset{\beta}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}_{\text{ridge}}(\mathbf{X})\beta \right\|_2^2 + \lambda^*_{\text{ridge}} \sum_{k=1}^{p^{(1)}_{\text{ridge}}} w_{\text{ridge},k}|\beta_k|,$$

$$\hat{\beta}^{(1)}_{\text{OLS}} = \underset{\beta}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}_{\text{OLS}}(\mathbf{X})\beta \right\|_2^2 + \lambda^*_{\text{OLS}} \sum_{k=1}^{p^{(1)}_{\text{OLS}}} w_{\text{OLS},k}|\beta_k|$$

▷ Apply threshold values

$\eta = [10^{-8}, 10^{-7}, \ldots, 10^{1}]$;

for $i = 1, \ldots,$ **card**(η) **do**

▷ Ridge and OLS regression estimates after variable selection

$$\hat{\beta}^{\text{OLS}}_{\text{ridge}}[i] = \underset{\beta_{\mathcal{K}_i}}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}_{\text{ridge},\mathcal{K}_i}(\mathbf{X})\beta_{\mathcal{K}_i} \right\|_2^2 \text{ where } \mathcal{K}_i = \{k : |\hat{\beta}^{(1)}_{\text{ridge},k}| \geq \eta_i\},$$

$$\hat{\beta}^{\text{OLS}}_{\text{OLS}}[i] = \underset{\beta_{\mathcal{K}_i}}{\text{argmin}} \left\| \dot{\mathbf{x}}_j - \boldsymbol{\Theta}^{(1)}_{\text{OLS},\mathcal{K}_i}(\mathbf{X})\beta_{\mathcal{K}_i} \right\|_2^2 \text{ where } \mathcal{K}_i = \{k : |\hat{\beta}^{(1)}_{\text{OLS},k}| \geq \eta_i\},$$

$$\text{BIC}_{\text{ridge},i} = \text{BIC}(\hat{\beta}^{\text{OLS}}_{\text{ridge}}[i])$$

$$\text{BIC}_{\text{OLS},i} = \text{BIC}(\hat{\beta}^{\text{OLS}}_{\text{OLS}}[i])$$

end for

$$\hat{\beta} = \left\{ \hat{\beta}^{\text{OLS}}[i], \hat{\beta}^{\text{ridge}}[i] \middle| i : \text{argmin}(\text{BIC}_{\text{OLS},i}, \text{BIC}_{\text{ridge},i}) \right\}$$

▷ STEP SIX: Bayesian regression and confidence intervals

perform Bayesian regression (BR) with $\hat{\beta}$

select final model: $\hat{\beta}_k \in \left[\hat{\beta}_k^{\text{BR}\{CI_{\text{lo}}\}}, \hat{\beta}_k^{\text{BR}\{CI_{\text{up}}\}} \right]$, and $0 < \hat{\beta}_k^{\text{BR}\{CI_{\text{lo}}\}}$ or $0 > \hat{\beta}_k^{\text{BR}\{CI_{\text{up}}\}}$

discovery of two- and three-dimensional nonlinear systems from limited observations and varying noise levels. Moreover, the recently developed Ensemble-SINDy is used to compare the success rates of the proposed approach. In this comparison, Ensemble-SINDy is deployed using the default bragging method, which obtains the median coefficient over all models [132].

5.2.2 Success Rates for Three-dimensional Systems

As previously discussed, many real-world applications involve three-dimensional systems to describe intricate dynamics in complex phenomena that cannot be accurately modelled using only two dimensions. Therefore, this study focuses primarily on the effectiveness of the proposed ARGOS-BI for discovering three-dimensional systems.

To illustrate the effectiveness of ARGOS-BI, Figure 5.1 shows its performance when discovering each system from data, highlighting model discovery accuracy above 80%. ARGOS-BI significantly outperformed Ensemble-SINDy across the examined systems, often requiring fewer observations and tolerating higher levels of noise when identifying the underlying dynamics automatically. Additionally, the method successfully discovered the trigonometric Thomas system from moderately-sized data sets, showcasing its ability to extract the true governing terms when additional multicollinearity exists in the design matrix.

To provide more insight into the decision-making process of ARGOS-BI, Figure 5.2 displays the uncertainty levels for each selected term in ARGOS-BI and provides a clear interpretation of the approach. The figure displays narrow credible intervals for each selected term in the equations of the Lorenz system. Furthermore, the figure shows that Bayesian inference enabled the removal of predictors whose uncertainty intervals contained zero while also estimating the standard deviation of the residuals from the regression model (σ).

Figure 5.3 demonstrates the method's ability to effectively remove terms from its prediction model since the z variable in the $\dot{x}_2$ equation crosses zero. The plot exhibits the posterior probability distribution crossing zero, indicating that the uncertainty levels were too high for the term to be included in the prediction model. Ultimately, this approach enabled the method to effectively select or remove terms

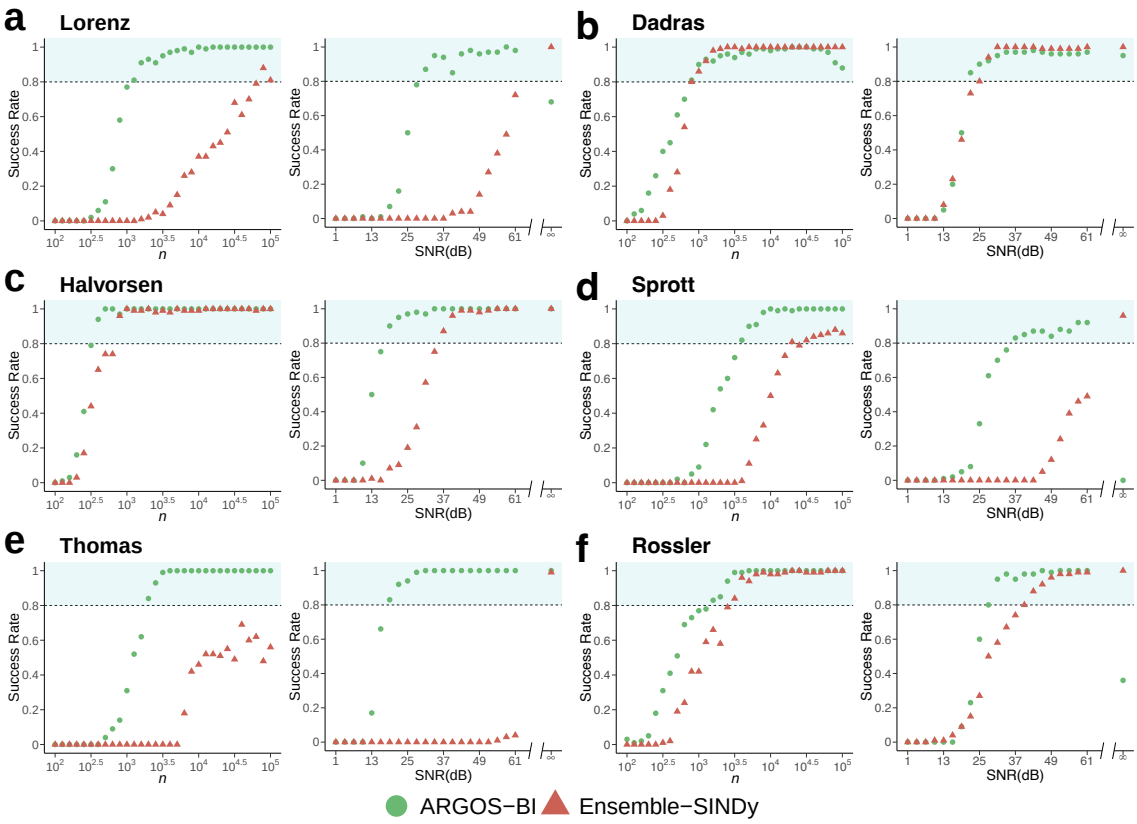

Figure 5.1: **Success rates of ARGOS-BI and Ensemble-SINDy for three-dimensional systems** The time-series length n is increased in the system while holding SNR = 49 dB (left panels) and fix $n = 5000$ when increasing the SNR (right panels). Nonlinear chaotic systems are shown in panels (**a**)-(**f**). Success rates are defined by the proportion of correctly discovered models for each system at each value of n and SNR. Shaded regions represent model discovery above 80%.

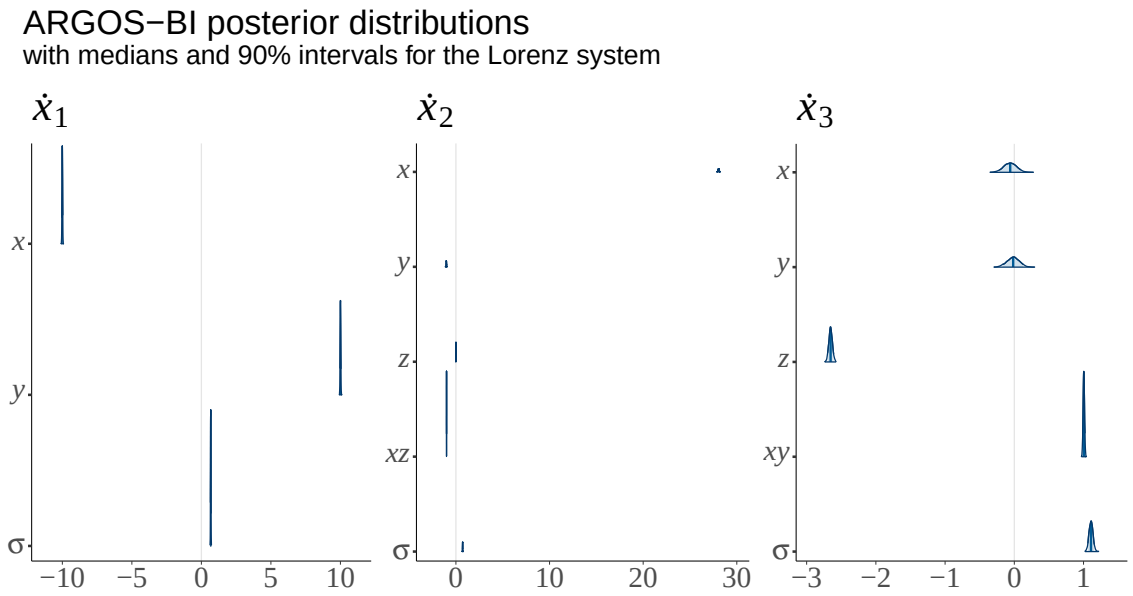

Figure 5.2: **Posterior Distributions for the Lorenz system.** Credible intervals for the identified variables of each equation were developed using the 90% confidence level. Terms with intervals containing zero are removed from the final prediction model.

from its prediction model and more accurately represent the underlying system.

As shown in Figure 5.1, the identification success rate of ARGOS-BI slightly reduced as n increased. These results can be explained by the number of influential observations in the data, which negatively impact the proposed method's efficacy in model discovery, as shown in Figure 5.4. Although the method identified the system accurately when $n = 10^4$, the data contained more influential data points that impacted the regression model with $n = 10^5$ observations, causing the approach to incorrectly select the intercept term in the $\dot{x}_1$ equation. Interestingly, this contrasts with the consistent identification demonstrated by the original ARGOS method, as reported in Chapter 3, which was seemingly unaffected by these observations.

Figure 5.4 highlights the Cook's Distance values for each observation, a critical statistical metric assessing the influence of individual data points on the overall model fit. A high Cook's Distance value suggests a significant influence of that observation on the model, potentially leading to a skew in the results. By excluding observations with Cook's Distance values exceeding 1, a measure of the influence of an observation on the overall model fit [143], this issue was mitigated, and the

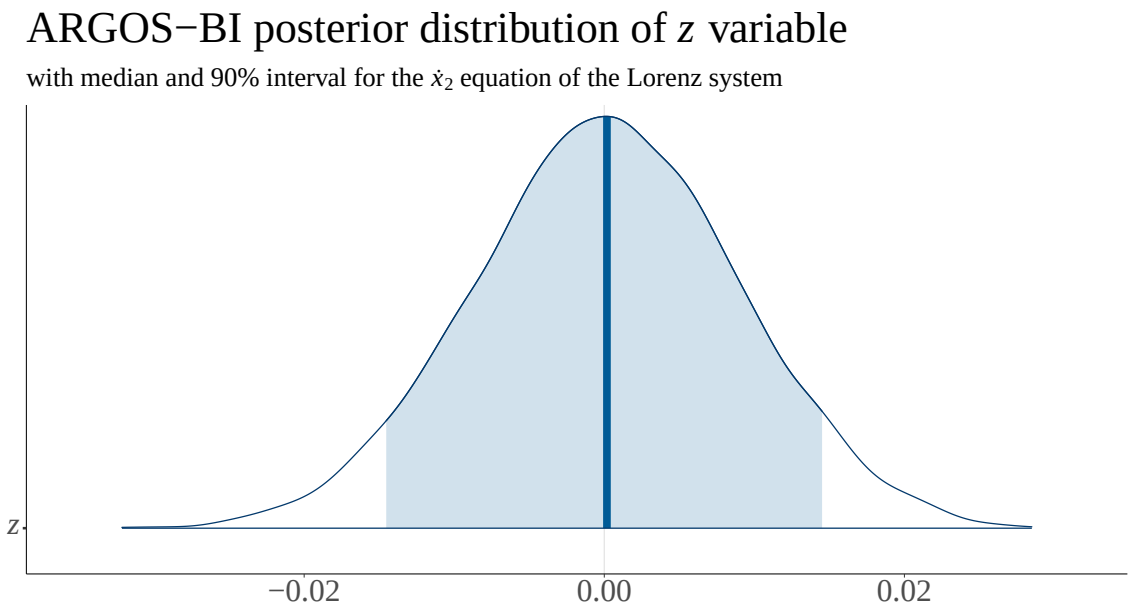

Figure 5.3: **Posterior Distribution for z variable in the $\dot{x}_2$ Lorenz system equation.** This figure demonstrates the rationale behind removing terms with credible intervals containing zero. The posterior distribution's median value indicates that uncertainty levels cross zero, suggesting this term should not be selected for prediction.

proposed method was able to identify the underlying equations more accurately. Thus, assessing these influential points suggests the identification results improve when adhering to the assumptions of linear regression.

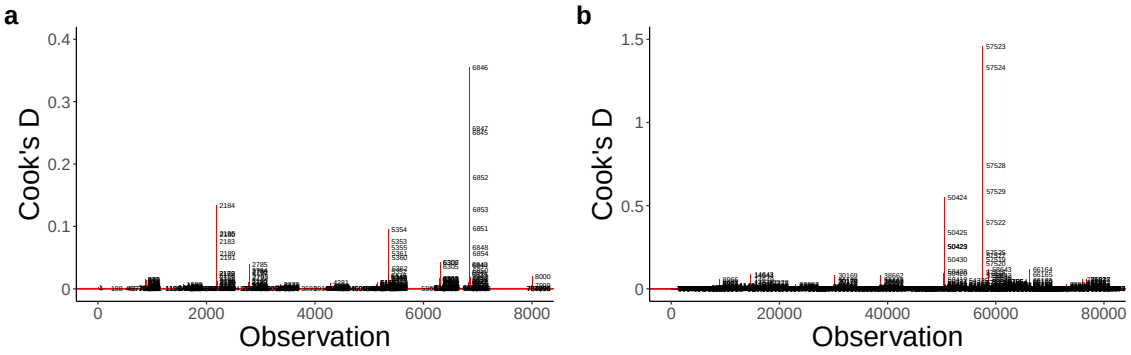

Figure 5.4: **Cook's Distance values for the identified model of the $\dot{x}_1$ equation of Dadras system.** These plots, corresponding to (**a**) $n = 10^4$ and (**b**) 10^5 observations, use Cook's Distance to highlight influential data points affecting the model's estimates. High Cook's Distance values signal potentially significant impacts on the model from specific observations, aiding in identifying outliers or pivotal data points at varying observation scales.

Similar to ARGOS in Chapter 3, when SNR = ∞, the proposed method's performance deteriorated for several systems, as shown in Figure 5.1. This decrease in identification accuracy occurred from the identified model's violation of the homoscedasticity assumption in linear regression, stating that residuals should exhibit a non-constant variance. As shown in Figure 5.5, the model identified by ARGOS-BI did not satisfy this assumption for the $\dot{x}_1$ equation of the Sprott system, subsequently leading to the addition of erroneous terms to alleviate this issue and reduce the heteroscedasticity in the residuals. However, as the noise in the system slightly increased, the variance among the residuals became more homoscedastic, enabling the proposed method to identify the system correctly. Therefore, ARGOS-BI also provides a practical alternative for accurately discovering the correct terms of the governing equations when data contain low levels of noise in the signal, which is helpful for many engineers working with real-world data.

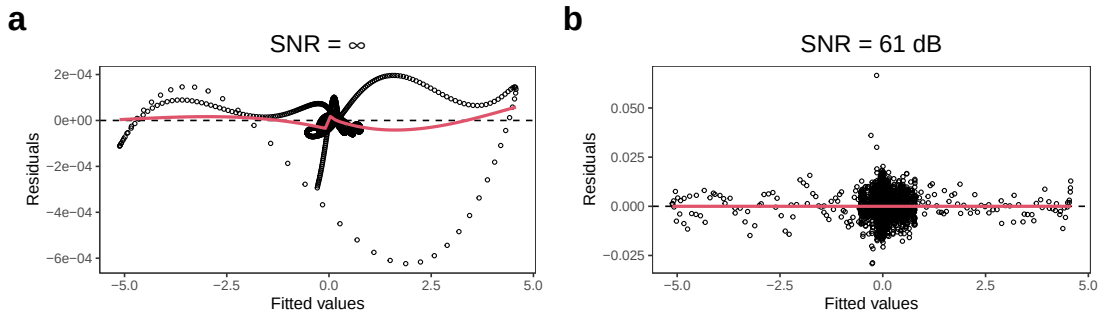

Figure 5.5: **Residuals vs fitted diagnostics for the identified model of the Sprott $\dot{x}_1$ equation**. Comparison of residuals for the prediction models identified for the Sprott system's $\dot{x}_1$ equation when data are (**a**) noiseless and (**b**) contaminated by SNR = 61 dB.

5.2.3 Success Rates for Two-dimensional Systems

Although the main focus of this chapter is to illustrate the proposed method's performance with three-dimensional systems, two-dimensional systems have been widely studied in system identification and control due to their simplicity and ease of visualisation. Figure 5.6 highlights the effectiveness of ARGOS-BI in identifying two-dimensional systems, showing that it consistently required fewer observations and tolerated higher levels of noise for these simpler dynamics. As opposed to previously

studied methods, the Bayesian approach enabled ARGOS-BI to achieve a 20% success rate when $n = 100$ for the two-dimensional oscillator with linear dynamics and the Lotka-Volterra system, indicating that the credible confidence intervals were effective in identifying the underlying equations.

As n and SNR increased, the proposed approach continued to improve in model discovery for the examined two-dimensional systems. These results can be attributed to the adaptive lasso's ability to more easily determine the correct equations for these simpler systems, enabling the Bayesian regression approach to develop simpler models and more accurate predictions. Hence, this approach proved more accurate and efficient for model discovery in many cases compared to Ensemble-SINDy, which struggled to extract the underlying equations for several systems.

Ultimately, these two-dimensional examples provide insight into the versatility of ARGOS-BI, outperforming Ensemble-SINDy in identifying the underlying equations for most systems tested here. Furthermore, the results show that ARGOS-BI provides an effective method for automatic model discovery of both two- and three-dimensional systems from data.

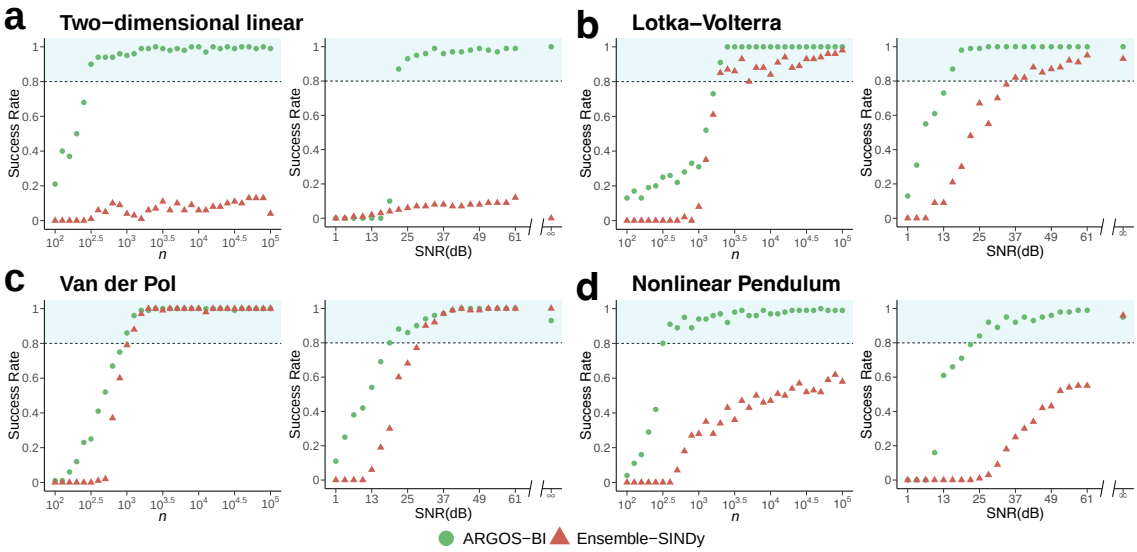

Figure 5.6: **Success rates of ARGOS-BI and Ensemble-SINDy for two-dimensional systems** The time-series length n is increased in the system while holding SNR = 49 dB (left panels) and fix $n = 5000$ when increasing the SNR (right panels). Panel (**a**) provides a linear system, while two-dimensional nonlinear systems are examined in (**b**)-(**d**). Success rates are defined by the proportion of correctly discovered models for each system at each value of n and SNR. Shaded regions represent model discovery above 80%.

5.2.4 Computational Time for ARGOS-BI

Figure 5.7 compares the computing time required for ARGOS-BI and Ensemble-SINDy [132]. The figure illustrates the time complexity (in seconds) required to perform each method using one CPU core with a single thread for the Lorenz system. Furthermore, the figure shows a wider variability in the proposed method's computational effort with lower values of n, which is consistent with the drawbacks of the approach in that it requires enough observations to develop valid estimates of the system. However, as n increased, the computational effort of each algorithm eventually began to converge, while ARGOS-BI more consistently identified the underlying system.

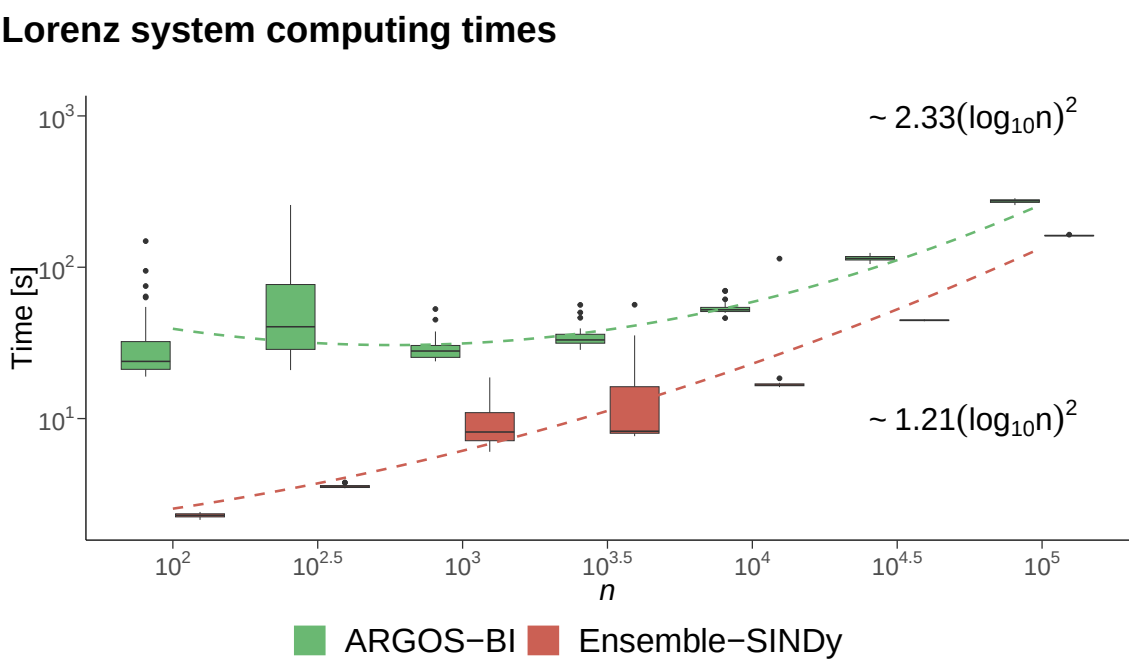

Figure 5.7: **Time-complexity (seconds) between ARGOS-BI and Ensemble-SINDy**. Boxplots depict the computational time required for model discovery over 100 instances for the Lorenz system. The black bar within each box represents the median computational time. Whiskers extending from each box show 1.5 times the interquartile range. Data points beyond the end of the whiskers are outlying points. Equations accompanying the dashed lines indicate the fitted mean computational time for each algorithm at various values of n.

Since determining the grid of window lengths, denoted as w, requires a constant that cannot be dropped, the optimal Savitzky-Golay filtering method can be described as $\mathcal{O}(nw)$, where n represents the number of observations. Moreover, the

computational complexity of the adaptive lasso is $\mathcal{O}(p^3 + np^2)$, where p denotes the number of variables in the design matrix. However, it is important to note that the adaptive lasso performs this process twice with ridge regression, meaning engineers should be aware of the slight increase in efficiency [81]. Furthermore, the computational complexity of Bayesian regression is $\mathcal{O}(p^3 + np^2)$. By transforming the algorithm into $\mathcal{O}(2 * (p^3 + np^2))$, the constant can be dropped, allowing ARGOS-BI to mimic the computational efficiency of the standard lasso method [134]. However, engineers should be aware that the multistage nature of the ARGOS-BI algorithm might necessitate more effort for accurately identifying the underlying equations.

5.3 Discussion

This chapter presented ARGOS-BI as an innovative extension to the ARGOS framework, enhancing the method's capability for automated system identification under noisy conditions. Through the integration of Bayesian inference, ARGOS-BI efficiently facilitates the discovery of governing equations from inherently noisy data.

Despite its promising contribution, the proposed method does have limitations. First, ARGOS-BI displayed a wider range of computational time required for model discovery with fewer observations than when the system contained sufficient data. In such cases, the adaptive lasso may fail to identify a sparse representation of the system, resulting in a more computationally expensive procedure during the MCMC process for estimating posterior probability values. However, as the length of the time series expanded, the adaptive lasso identified a more accurate, sparse representation of the system and facilitated the Bayesian regression approach to develop a more accurate identification of the true model.

When applying regression-based algorithms, such as the original ARGOS, ASINDy, or the proposed ARGOS-BI, attention must be paid to the potential impact of influential observations on the model's accuracy. While ARGOS' demonstrated consistent identification with an increase in observations, as seen in Chapter 3, ARGOS-BI's success rate slightly decreased with the expansion of the Dadras system. This reduction in performance can be attributed to potential outliers in the

data, which were identifiable using Cook's Distance. These observations can be excluded during regression by determining influential data points that might disproportionately affect the model's performance, enabling the proposed approach to select the true model accurately.

ARGOS-BI's performance was also affected by the violation of homoscedasticity, similar to ARGOS's behaviour in Chapter 3. The residuals displayed non-constant variance for deterministic systems, and the proposed approach identified extraneous terms to improve prediction accuracy while sacrificing correct identification. However, this issue was again mitigated with a slight increase in noise in the system, and ARGOS-BI could represent each system more accurately.

The proposed method only assumes a parsimonious underlying model structure and employs independent and identically distributed Gaussian prior probabilities for each term in its prediction model. This choice is strategically sound for a broad range of applications due to the balance it offers between computational efficiency and accuracy. Gaussian priors provide a mathematical simplicity that facilitates quicker calculations and generally acceptable accuracy, making them a preferred choice in many engineering contexts. However, the decision to employ specific priors should not be taken lightly, especially in real-world settings where the underlying dynamics might be complex or not well-understood.

While Gaussian priors are generally robust and versatile, different scenarios might necessitate alternative distributions. For instance, employing a Laplace prior can promote sparsity in the final model, which is advantageous in systems where only a few predictors are truly influential. Alternatively, using a Cauchy distribution can enhance the model's robustness against influential observations, reducing the impact of outliers.

It's crucial to note that these alternative priors come with their own challenges. They often require more intricate hyperparameter tuning, which can introduce a higher computational cost and complexity. Such complexity not only demands more from the computational infrastructure but also from the users in terms of their expertise and understanding of Bayesian statistics. For example, in practice, the efficiency of a Laplace prior in promoting sparsity might be offset by the increased difficulty

in estimating the correct scale parameter, making it less accessible for engineers without in-depth statistical training. Therefore, while exploring different prior distributions can theoretically enhance the model's ability to identify the underlying system accurately, one must balance this potential gain against increased complexity and the risk of model misspecification. The strength of the Gaussian prior lies in its general applicability and tractability, especially when dealing with large data sets where the central limit theorem reinforces its suitability. Thus, although integrating different prior distributions can be beneficial in refining system discovery and prediction accuracy, engineers are advised to weigh these benefits against the original method's simplicity and efficiency. Careful consideration is needed to ensure the chosen prior is appropriate for the specific context and does not inadvertently compromise the model's overall utility.

In summary, ARGOS-BI offers a valuable contribution to the field of system identification by providing a statistically rigorous and noise-robust method for automating model discovery. Its Bayesian approach holds great potential for advancing the state-of-the-art and paving the way for new and innovative methods for extracting governing equations from data. By leveraging the benefits of Bayesian regression and developing credible intervals for model inference, this approach provides a statistically rigorous and noise-robust identification procedure for ARGOS. As a result, researchers and engineers can employ ARGOS-BI to effectively discover the governing equations of dynamical systems from noisy data while providing uncertainty measures in the process.

5.4 Summary

- ARGOS-BI, a Bayesian inference approach, provides a probabilistic method for automating system identification, producing uncertainty measurements and optimal system representation through credible intervals.
- The method establishes a computationally efficient approach for identifying systems automatically and consistently outperforms Ensemble-SINDy in model discovery, despite varying computational times for smaller data sets.
- Results emphasise the importance of satisfying linear regression assumptions, demonstrating better performance when data is free from significant outliers, and improved identification when slight levels of noise are present, as opposed to deterministic systems.
- The proposed approach is an efficient alternative to frequentist methods, providing reliable results and noise-robust identification capabilities.

CHAPTER 6

Conclusion

This thesis has presented innovative contributions to data-driven system identification, focusing on the automated discovery of interpretable prediction models. The initial chapters underlined the potential of data-centric engineering across a broad spectrum of applications, laying the groundwork for developing advanced statistical and machine-learning methods to perform system identification.

Chapter 3 introduced ARGOS, an automatic method for identifying systems of ODEs from data. ARGOS combines signal denoising, sparse regression, and bootstrap sampling to formulate a reliable solution path for model discovery. A comparison with the recently developed SINDy with AIC [67] demonstrated ARGOS's efficiency in automating model discovery.

The subsequent chapter presented an enhancement to the original SINDy algorithm, ASINDy, that integrates standard unsupervised learning methods and establishes a grid of thresholds to perform the more common model evaluation approach, K-fold cross-validation. By employing a range of threshold values for the sequential thresholding algorithm, ASINDy outperforms the original SINDy method, offering an approach that reduces the need for engineers to determine sparsity-promoting hyperparameters manually.

Finally, Chapter 5 extended the ARGOS framework by proposing a Bayesian approach as an alternative to the frequentist bootstrap sampling technique employed in Chapter 3. Despite the method's limitations, the proposed Bayesian implementation of ARGOS illustrated its consistency and efficacy in automating model discovery compared to Ensemble-SINDy [132].

An additional contribution of this thesis is an optimal Savitzky-Golay filtering method. This optimisation enhances the ability of each algorithm to automatically smooth each vector of the state matrix and develop the numerical derivative, a critical element in many data-driven fields. Without this approach, users are left to determine the Savitzky-Golay algorithm's parameters manually.

6.1 Implications and Applications

The methods developed in this thesis hold significant potential implications for diverse fields. In engineering, these algorithms can streamline the process of model discovery for systems of ordinary differential equations found in climate modelling or intricate manufacturing processes, enabling more efficient development of procedures for optimisation, control, and analysis. Throughout biological sciences, these methods can aid in examining systems like cell signalling or ecological dynamics, potentially offering profound insights into complex phenomena. Moreover, economists leverage these algorithms to discover equations that describe historical performance data, modelling revenue, cash flow, expenses, or sales, and improve the accuracy of financial forecasts. Finally, astrophysicists are another group that could benefit from data-driven system identification, as building cosmological models from large amounts of data often requires a sparse representation in the form of ordinary differential equations. These are just a few examples of potential real-world applications, and the methodology presented here serves as a stepping stone towards more reliable, accurate, and automated modelling of dynamical systems, hopefully propelling scientific research and technological advancements across various disciplines.

6.2 Future Directions

The methods developed in this thesis offer noise-resistant system identification for equations, which is crucial for engineers when extracting valuable information from data. However, some statisticians may question the post-selection inference methods avoided here [144]. Despite the accurate model assessment techniques used in this thesis, the field of post-selection inference for model selection is still developing. While integrating these methods into ARGOS may yield a more rigorous framework, their current state of development prevented their application here as the statistical community continues seeking secure implementation procedures. Consequently, the techniques discussed in this thesis provide a statistically robust approach to system identification.

The importance of balancing computational efficiency and statistical inference cannot be stressed enough. This thesis introduced three methods that automate the model discovery of dynamical systems. When dealing with large-scale data ($n = 10^5$), the bootstrap sampling employed by the original ARGOS framework requires significant time to perform model discovery compared to ASINDy and ARGOS-BI. However, the original ARGOS method displayed a smaller increase in computational effort as n increased. Therefore, although ARGOS did not necessarily take less time to accomplish model discovery, the results suggested that it may become more efficient with larger data sets.

Moreover, substituting bootstrap sampling with Bayesian regression sacrificed a slight degree of accuracy but considerably increased efficiency with fewer observations. These results showed the flexibility and enduring success of the ARGOS framework for system identification. Although computational efficiency is an important metric, emphasis must also be placed on each algorithm's success rate. Therefore, engineers should consider the MSE of each method's predictions when assessing their final model.

The systematic analysis presented in this work provides a practical evaluation of identification procedures, emphasising their advantages and drawbacks while encouraging comparisons between studies. A standardised framework facilitates a more relevant assessment, improving transparency, replication, and scientific rigour. This

approach addresses knowledge gaps and encourages interdisciplinary collaboration, fostering a unified scientific community.

Automating system identification has the potential to expedite discoveries and enhance accuracy across various scientific disciplines, from engineering, chemistry and biology, to economics and physics. With these methods, engineers can generate large models with numerous variables that would otherwise be daunting and labour-intensive to create manually. Additionally, the proposed algorithms minimise the risk of errors and inconsistencies often associated with manual model development, enhancing model accuracy and reliability and leading to more effective decision-making and more robust engineering designs. In conclusion, methods for automatically discovering dynamical systems from data are essential for advancing scientific research and accelerating technological progress.

Bibliography

[1] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine learning in materials informatics: Recent applications and prospects," *npj Computational Materials*, vol. 3, no. 1, p. 54, Dec. 2017. 1.1

[2] S. Hiemer and S. Zapperi, "From mechanism-based to data-driven approaches in materials science," *Materials Theory*, vol. 5, no. 1, p. 4, Sep. 2021.

[3] Y. Xie, M. Ebad Sichani, J. E. Padgett, and R. DesRoches, "The promise of implementing machine learning in earthquake engineering: A state-of-the-art review," *Earthquake Spectra*, vol. 36, no. 4, pp. 1769–1801, Nov. 2020.

[4] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, M. Z. Bazant, S. J. Harris, W. C. Chueh, and R. D. Braatz, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, May 2019. 1.1

[5] J. Edghill and D. Towill, "The use of system dynamics in manufacturing systems engineering," *Transactions of the Institute of Measurement and Control*, vol. 11, no. 4, pp. 208–216, Nov. 1989. 1.1

[6] P. Denno, C. Dickerson, and J. A. Harding, "Dynamic production system identification for smart manufacturing systems," *Journal of Manufacturing Systems*, vol. 48, pp. 192–203, Jul. 2018.

[7] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *Journal of Manufacturing Systems*, vol. 48, pp. 78–86, Jul. 2018. 1.1

[8] B. F. Spencer, V. Hoskere, and Y. Narazaki, "Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, Apr. 2019. 1.1

[9] M. Karimi-Ghartemani and M. Iravani, "A method for synchronization of power electronic converters in polluted and variable-frequency environments," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1263–1270, Aug. 2004. 1.1

[10] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004. 1.1

[11] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Oct. 2000, pp. 153–158.

[12] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*, ser. Control Engineering. Birkhäuser New York, NY, 2014.

[13] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, Apr. 2009.

[14] A. Babin, "Data-driven system identification and optimal control of an active rotor-bearing system," *IOP Conference Series: Materials Science and Engineering*, vol. 1047, no. 1, p. 012053, Feb. 2021.

[15] J. Ching, J. L. Beck, and K. A. Porter, "Bayesian state and parameter estimation of uncertain dynamical systems," *Probabilistic Engineering Mechanics*, vol. 21, no. 1, pp. 81–96, Jan. 2006. 1.1

[16] K. Worden, C. R. Farrar, J. Haywood, and M. Todd, "A review of nonlinear dynamics applications to structural health monitoring," *Structural Control and Health Monitoring*, vol. 15, no. 4, pp. 540–567, 2008. 1.1

[17] J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88, no. 9, pp. 3079–3087, Sep. 2011. 1.1

[18] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 1st ed. New York: Cambridge University Press, Jan. 2019. 1.1, 2.6, 2.6, 2.9, 2.11

[19] C. Rhie and W. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *American Institute of Aeronautics and Astronautics Journal*, vol. 21, no. 11, pp. 1525–1532, 1983. 1.1

[20] S. L. Brunton, T. Duriez, and B. R. Noack, *Machine Learning Control - Taming Nonlinear Dynamics and Turbulence*, ser. Fluid Mechanics and Its Applications. Springer Cham, 2017, vol. 116.

[21] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a Time Series," *Physical Review Letters*, vol. 45, no. 9, pp. 712–716, Sep. 1980. 1.1

[22] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, Feb. 2013. 1.1

[23] O. Sporns, G. Tononi, and G. Edelman, "Theoretical Neuroanatomy: Relating Anatomical and Functional Connectivity in Graphs and Cortical Connection Matrices," *Cerebral Cortex*, vol. 10, no. 2, pp. 127–141, Feb. 2000.

[24] J. C. Reijneveld, S. C. Ponten, H. W. Berendse, and C. J. Stam, "The application of graph theoretical analysis to complex networks in the brain," *Clinical Neurophysiology*, vol. 118, no. 11, pp. 2317–2331, Nov. 2007. 1.1

[25] Á. Cartea and D. del Castillo Negrete, "Fractional diffusion models of option prices in markets with jumps," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 2, pp. 749–763, Feb. 2007. 1.1

[26] X. Y. Zhou and G. Yin, "Markowitz's Mean-Variance Portfolio Selection with Regime Switching: A Continuous-Time Model," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1466–1482, Jan. 2003.

[27] Q. Zhang, "Stock Trading: An Optimal Selling Rule," *SIAM Journal on Control and Optimization*, vol. 40, no. 1, pp. 64–87, Jan. 2001. 1.1

[28] "Mark Girolami appointed to lead The Alan Turing Institute-Lloyd's Register Foundation data-centric engineering programme," https://www.turing.ac.uk/news/mark-girolami-appointed-lead-alan-turing-institute-lloyds-register-foundation-data-centric, Feb. 2017. 1.1

[29] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: A survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, Aug. 2018. 1.1

[30] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, Mar. 2019.

[31] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter Cities and Their Innovation Challenges," *Computer*, vol. 44, no. 6, pp. 32–39, Jun. 2011. 1.1

[32] A. Bazzani, B. Giorgini, S. Rambaldi, R. Gallotti, and L. Giovannini, "Statistical laws in urban mobility from microscopic GPS data in the area of Florence," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 05, p. P05001, May 2010. 1.1

[33] S. Dodge, "A Data Science Framework for Movement," *Geographical Analysis*, vol. 53, no. 1, pp. 92–112, 2021. 1.1

[34] A. Parisio, E. Rikos, and L. Glielmo, "A Model Predictive Control Approach to Microgrid Operation Optimization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1813–1827, Sep. 2014. 1.1

[35] H.-x. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, Aug. 2012.

[36] S. A. Kalogirou, "Artificial neural networks in renewable energy systems applications: A review," *Renewable and Sustainable Energy Reviews*, vol. 5, no. 4, pp. 373–401, Dec. 2001. 1.1

[37] B. Wett and W. Rauch, "The role of inorganic carbon limitation in biological nitrogen removal of extremely ammonia concentrated wastewater," *Water Research*, vol. 37, no. 5, pp. 1100–1110, Mar. 2003. 1.1

[38] R. Bürger, S. Diehl, and I. Nopens, "A consistent modelling methodology for secondary settling tanks in wastewater treatment," *Water Research*, vol. 45, no. 6, pp. 2247–2260, Mar. 2011.

[39] D. W. Porter, B. P. Gibbs, W. F. Jones, P. S. Huyakorn, L. L. Hamm, and G. P. Flach, "Data fusion modeling for groundwater systems," *Journal of Contaminant Hydrology*, vol. 42, no. 2, pp. 303–335, Mar. 2000. 1.1

[40] S. Vardoulakis, B. E. A. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: A review," *Atmospheric Environment*, vol. 37, no. 2, pp. 155–182, Jan. 2003. 1.1

[41] R. A. Zaveri, R. C. Easter, J. D. Fast, and L. K. Peters, "Model for Simulating Aerosol Interactions and Chemistry (MOSAIC)," *Journal of Geophysical Research: Atmospheres*, vol. 113, no. D13, 2008.

[42] F. S. Binkowski and S. J. Roselle, "Models-3 Community Multiscale Air Quality (CMAQ) model aerosol component 1. Model description," *Journal of Geophysical Research: Atmospheres*, vol. 108, no. D6, 2003. 1.1

[43] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Jan. 2005. 1.1

[44] X. Hong, R. Mitchell, S. Chen, C. Harris, K. Li, and G. Irwin, "Model selection approaches for non-linear system identification: A review," *International Journal of Systems Science*, vol. 39, no. 10, pp. 925–946, Oct. 2008. 1.1

[45] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall PTR, 1999. 1.1, 2.3.3

[46] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016. 1.1, 2.6, 2.6, 2.6, 2.6, 2.6, 2.9, 2.9, 2.10, 2.11, 3.1.1, 3.2.1, 3.4.1, 3.4.1, 3.4.2, 3.4.2, 3.4.2, 4.1.1, 4.2.1, 4.5, 4.3, 4.16, 4.5, A.6

[47] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, p. e1602614, Apr. 2017. 2.6, 2.9, 2.9, 2.10, 4.5, 4.16

[48] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 357, pp. 125–141, Mar. 2018. 1.1, 2.10

[49] M. Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations," *Journal of Machine Learning Research*, vol. 19, no. 25, pp. 1–24, 2018. 1.1

[50] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed., ser. Springer Texts in Statistics. New York, NY: Springer, 2021, vol. 103. 1.1, 1.4, 1.4, 2.1.1, 2.1.1, 2.1.1, 2.1.1, 2.1.1, 2.1.1, 2.1.1, 2.1.2, 2.2.1, 2.2.2, 2.2.2, 2.2.4, 2.2.4, 2.2.4, 2.2.4, 2.3.1, 2.3.1, 2.3.2, 2.3.2, 2.3.3, 2.5, 2.5.1, 2.5.1, 2.5.1, 2.5.2, 2.11, 4.1.1

[51] R. Tibshirani, J. H. Friedman, and T. Hastie, *The Elements of Statistical Learning*. New York: Springer, 2009. 1.4, 2.2.1, 2.2.1, 2.2.4, 2.2.4, 2.2.4, 2.2.4, 2.2.5, 2.3.1, 2.3.2, 2.3.2, 2.3.2, 2.3.3, 2.5.1, 2.11, 3.1.2, 3.2.1

[52] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015. 1.1, 1.4, 2.2.1, 2.2.4, 2.2.4, 2.2.4, 2.2.4, 2.2.4, 2.2.5, 2.11, 3.1.1, 3.2.1

[53] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, Apr. 2009. 1.1, 2.6

[54] B. Daniels and I. Nemenman, "Automated adaptive inference of phenomenological dynamical models," *Nature Communications*, vol. 6, no. 1, p. 8133, Nov. 2015.

[55] J. L. Callaham, J. V. Koch, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Learning dominant physical processes with data-driven balance models," *Nature Communications*, vol. 12, p. 1016, Dec. 2021.

[56] S.-M. Udrescu and M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, p. eaay2631, Apr. 2020. 1.1, 2.6

[57] H. Schaeffer, G. Tran, and R. Ward, "Extracting sparse high-dimensional dynamics from limited data," *SIAM Journal on Applied Mathematics*, vol. 78, no. 6, pp. 3279–3295, Jan. 2018. 1.1

[58] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes," *Journal of Computational Physics*, vol. 348, pp. 683–693, Nov. 2017.

[59] R. Guimerà, I. Reichardt, A. Aguilar-Mogas, F. A. Massucci, M. Miranda, J. Pallarès, and M. Sales-Pardo, "A Bayesian machine scientist to aid in the solution of challenging scientific problems," *Science Advances*, vol. 6, no. 5, p. 107077, Jan. 2020. 1.1

[60] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, p. 4950, Dec. 2018. 1.1, 2.6

[61] S. A. Kalogirou, "Applications of artificial neural-networks for energy systems," *Applied Energy*, vol. 67, no. 1, pp. 17–35, Sep. 2000. 1.1

[62] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, "Predicting catastrophes in nonlinear dynamical systems by compressive sensing," *Physical Review Letters*, vol. 106, no. 15, p. 154101, Apr. 2011. 1.1, 2.6

[63] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, "A Unified Sparse Optimization Framework to Learn Parsimonious Physics-Informed Models From Data," *IEEE Access*, vol. 8, pp. 169 259–169 271, 2020. 1.1, 2.10

[64] A. Cortiella, K.-C. Park, and A. Doostan, "Sparse identification of nonlinear dynamical systems via reweighted ℓ_1-regularized least squares," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113620, Apr. 2021. 1.1, 2.6, 2.10, 3.4.3, 3.4.3, 3.4.3, 3.4.3, A.7, A.8, A.8

[65] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher, "Sparse dynamics for partial differential equations," *Proceedings of the National Academy of Sciences*, vol. 110, no. 17, pp. 6634–6639, Apr. 2013.

[66] H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2197, p. 20160446, Jan. 2017. 1.1

[67] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2204, p. 20170009, Aug. 2017. 1.1, 2.6, 2.9, 2.9, 2.10, 3, 3.2.2, 3.5, 6

[68] S. Zhang and G. Lin, "Robust data-driven discovery of governing physical laws with error bars," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2217, p. 20180305, Sep. 2018. 1.1, 2.10

[69] C. R. Shalizi, *Advanced Data Analysis from an Elementary Point of View*, 2019. 2.1.2, 2.1.2, 2.1.2, 2.1.2, 2.1.2

[70] A. Agresti, *Foundations of Linear and Generalized Linear Models*, 1st ed. Wiley, Feb. 2015. 2.1.2, 2.3.2, 2.3.2, 2.3.2

[71] S. Chatterjee and J. S. Simonoff, *Handbook of Regression Analysis With Applications in R*. Wiley, 2020. 2.1.2, 2.2.1, 2.2.2

[72] A. Tangirala, *Principles of System Identification: Theory and Practice*. CRC Press, 2018. 2.1.2, 2.6, 2.6

[73] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004. 2.2.1, 2.2.3, 2.2.3, 2.2.3, 2.2.3, 2.2.4, 2.2.4, 2.2.4

[74] S. Chatterjee and A. Hadi, *Regression Analysis by Example*. Wiley, 2012. 2.2.2, 2.2.2, 2.2.4

[75] G. Calafiore and L. El Ghaoui, *Optimization Models*. Cambridge: Cambridge University Press, 2014. 2.2.4, 2.2.4, 2.2.4, 2.2.4, 2.9

[76] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, pp. 267–288, 1996. 2.2.4, 2.2.4, 3.1.1

[77] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970. 2.2.4

[78] T. Hastie, J. Qian, and K. Tay, "Glmnet Vignette 2021," Nov. 2021. 2.2.4

[79] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *The Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, Jun. 2006. 2.2.4

[80] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via Coordinate Descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010. 2.2.4, 3.1.2

[81] H. Zou, "The Adaptive Lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006. 2.2.5, 2.2.5, 3.1.1, 3.2.2, 5.2.4

[82] R. Tibshirani and L. Wasserman, "Sparsity, the Lasso, and friends," 2017. 2.2.5

[83] J. Huang, S. Ma, and C.-H. Zhang, "Adaptive Lasso for Sparse High-Dimensional Regression Models," *Statistica Sinica*, vol. 18, pp. 1603–1618, 2008. 2.2.5

[84] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*, ser. Springer Series in Statistics. Springer Berlin, Heidelberg, 2011. 2.2.5, 2.2.5

[85] R. Li and H. Cui, "Variable Selection via Regularization," in *Encyclopedia of Environmetrics*. Wiley StatsRef: Statistics Reference Online, 2013, p. 9. 2.2.5

[86] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, Mar. 1978. 2.3.2, 3.1.1, 3.1.2

[87] H. Zou, T. Hastie, and R. Tibshirani, "On the "degrees of freedom" of the lasso," *The Annals of Statistics*, vol. 35, no. 5, Oct. 2007. 2.3.2

[88] J. Qiao, L. Wang, and C. Yang, "Adaptive lasso echo state network based on modified Bayesian information criterion for nonlinear system modeling," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6163–6177, Oct. 2019. 2.3.2

[89] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, ser. Monographs on Statistics and Applied Probability. New York: Chapman & Hall, 1993, no. 57. 2.3.3, 3.1.1, 3.1.2, 3.2.2

[90] A. M. Zoubir and D. R. Iskander, *Bootstrap Techniques for Signal Processing*. Cambridge: Cambridge University Press, 2004. 2.3.3, 2.11, 3.1.2

[91] M. L. Rizzo, *Statistical Computing with R*, 2nd ed. Boca Raton: CRC Press, Taylor & Francis Group, 2019. 2.3.3

[92] J. D. Lee, D. L. Sun, Y. Sun, and J. E. Taylor, "Exact post-selection inference, with application to the lasso," *The Annals of Statistics*, vol. 44, no. 3, Jun. 2016. 2.3.3

[93] R. Christensen, W. Johnson, A. Branscum, and T. E. Hanson, *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. Boca Raton: CRC Press, Jul. 2010. 2.4, 2.4, 2.4, 5.1

[94] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian Model Averaging: A Tutorial," *Statistical Science*, vol. 14, no. 4, pp. 382–401, 1999. 2.4

[95] A. Ciccone and M. Jarociński, "Determinants of Economic Growth: Will Data Tell?" *American Economic Journal: Macroeconomics*, vol. 2, no. 4, pp. 222–246, 2010. 2.4

[96] H. Best and C. Wolf, Eds., *The SAGE Handbook of Regression Analysis and Causal Inference*. Los Angeles [Calif.]: SAGE Reference, 2015. 2.4

[97] R. D. Morey, R. Hoekstra, J. N. Rouder, M. D. Lee, and E.-J. Wagenmakers, "The fallacy of placing confidence in confidence intervals," *Psychonomic Bulletin & Review*, vol. 23, no. 1, pp. 103–123, Feb. 2016. 2.4, 5.1

[98] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New York, NY: Pearson, 2018. 2.5.3

[99] S. L. Bangare, A. Dubal, P. S. Bangare, and S. Patil, "Reviewing Otsu's Method For Image Thresholding," *International Journal of Applied Engineering Research*, vol. 10, no. 9, pp. 21 777–21 783, May 2015. 2.5.3

[100] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. 2.5.3

[101] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York, NY: Springer Science & Business Media, 2013, vol. 42. 2.6, 4.4.4

[102] N. J. Higham and M. R. Dennis, Eds., *The Princeton Companion to Applied Mathematics*. Princeton: Princeton University Press, 2015. 2.6, 2.6

[103] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964. 2.6, 2.6, 2.8, 2.11, 3.1.1

[104] S. Rudy, A. Alla, S. L. Brunton, and J. N. Kutz, "Data-Driven Identification of Parametric Partial Differential Equations," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 2, pp. 643–660, Jan. 2019. 2.6, 2.10

[105] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 445–22 451, Nov. 2019. 2.6, 2.9

[106] M. Quade, M. Abel, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for rapid model recovery," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 6, p. 063116, Jun. 2018. 2.10

[107] B. M. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz, "Discovery of Physics From Data: Universal Laws and Discrepancies," *Frontiers in Artificial Intelligence*, vol. 3, p. 25, Apr. 2020. 2.6, 2.9

[108] P. Zhang, *Advanced Industrial Control Technology*. Amsterdam Boston Heidelberg: William Andrew, an imprint of Elsevier, 2010. 2.6

[109] W. E, "A Proposal on Machine Learning via Dynamical Systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, pp. 1–11, Mar. 2017. 2.6

[110] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 6, p. 061104, Jun. 2018.

[111] S. Pan and K. Duraisamy, "Long-Time Predictive Modeling of Nonlinear Dynamical Systems Using Neural Networks," *Complexity*, vol. 2018, pp. 1–26, Dec. 2018. 2.6

[112] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, pp. 9943–9948, Jun. 2007. 2.6

[113] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for nonlinear systems Part I: Deterministic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328, Feb. 1985. 2.6

[114] S. A. Billings and H. L. Wei, "The wavelet-NARMAX representation: A hybrid model structure combining polynomial models with multiresolution wavelet decompositions," *International Journal of Systems Science*, vol. 36, no. 3, pp. 137–152, Feb. 2005.

[115] G. Zito and I. Landau, "Narmax model identification of a variable geometry turbocharged diesel engine," in *Proceedings of the 2005, American Control Conference, 2005.* Portland, OR, USA: IEEE, 2005, pp. 1021–1026. 2.6

[116] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2219, p. 20180335, Nov. 2018. 2.6, 2.9, 2.9

[117] K. Kaheman, J. N. Kutz, and S. L. Brunton, "SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, no. 2242, p. 20200279, Oct. 2020.

[118] D. E. Shea, S. L. Brunton, and J. N. Kutz, "SINDy-BVP: Sparse identification of nonlinear dynamics for boundary value problems," *Physical Review Research*, vol. 3, no. 2, p. 023255, Jun. 2021.

[119] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, Jun. 2016. 2.6, 2.9, 2.9

[120] S. Chen, A. Shojaie, and D. M. Witten, "Network Reconstruction From High-Dimensional Ordinary Differential Equations," *Journal of the American Statistical Association*, vol. 112, no. 520, pp. 1697–1707, Oct. 2017. 2.6

[121] M. Moscoso, A. Novikov, G. Papanicolaou, and C. Tsogka, "The Noise Collector for sparse recovery in high dimensions," *Proceedings of the National Academy of Sciences*, vol. 117, no. 21, pp. 11 226–11 232, May 2020. 2.6

[122] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Boston, MA: Pearson, 2011. 2.7, 2.7, 2.7

[123] *Signal Processing Toolbox User's Guide.* Natwick, MA: The MathWorks, Inc., 2021. 2.7

[124] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York: Cambridge University Press, 2007. 2.8, 3.1.1

[125] R. Schafer, "What is a Savitzky-Golay Filter? [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, Jul. 2011. 2.8, 2.11

[126] M. Sadeghi, F. Behnia, and R. Amiri, "Window Selection of the Savitzky–Golay Filters for Signal Recovery From Noisy Measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5418–5427, Aug. 2020. 2.8

[127] J.-C. Loiseau and S. L. Brunton, "Constrained sparse Galerkin regression," *Journal of Fluid Mechanics*, vol. 838, pp. 42–67, Mar. 2018. 2.9

[128] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse Identification of Nonlinear Dynamics with Control (SINDYc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016. 2.9

[129] H. Schaeffer and S. G. McCalla, "Sparse model selection via integral terms," *Physical Review E*, vol. 96, no. 2, p. 023302, Aug. 2017. 2.9, 2.10

[130] G. Tran and R. Ward, "Exact Recovery of Chaotic Systems from Highly Corrupted Data," *Multiscale Modeling & Simulation*, vol. 15, no. 3, pp. 1108–1129, Jan. 2017. 2.9, 2.10, 3.4.2, A.5

[131] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nature Communications*, vol. 8, p. 19, Dec. 2017. 2.9

[132] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton, "Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 478, no. 2260, p. 20210904, Apr. 2022. 2.9, 5, 5.2.1, 5.2.4, 6

[133] O. Fajardo-Fontiveros, I. Reichardt, H. R. De Los Ríos, J. Duch, M. Sales-Pardo, and R. Guimerà, "Fundamental limits to learning closed-form mathematical models from data," *Nature Communications*, vol. 14, no. 1, p. 1043, Feb. 2023. 2.10

[134] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, Apr. 2004. 3.2.2, 5.2.4

[135] A. J. Lotka, "Contribution to the Theory of Periodic Reactions," *The Journal of Physical Chemistry*, vol. 14, no. 3, pp. 271–274, Mar. 1910. 3.4.2

[136] G. T. Naozuka, H. L. Rocha, R. S. Silva, and R. C. Almeida, "SINDy-SA framework: Enhancing nonlinear system identification with sensitivity analysis," *Nonlinear Dynamics*, vol. 110, no. 3, pp. 2589–2609, Nov. 2022. 3.4.2, 4.4.4, 4.4.4, A.4, A.11, A.11

[137] B. Van der Pol, "A theory of the amplitude of free and forced triode vibrations," *Radio Review (London)*, vol. 1, pp. 701–710, 1920. 3.4.3

[138] Y. Cai, X. Wang, G. Joós, and I. Kamwa, "An Online Data-Driven Method to Locate Forced Oscillation Sources From Power Plants Based on Sparse

Identification of Nonlinear Dynamics (SINDy)," *IEEE Transactions on Power Systems*, vol. 38, no. 3, pp. 2085–2099, May 2023. 3.4.4, 3.4.4, 3.4.4, 3.4.4, 3.4.4, 3.4.4, 3.4.4

[139] J. C. Sprott, *Chaos and Time-Series Analysis*. USA: Oxford University Press, Inc., 2003. 4.2.3

[140] J. C. Sprott and K. E. Chlouverakis, "Labyrinth chaos," *International Journal of Bifurcation and Chaos*, vol. 17, no. 06, pp. 2097–2108, Jun. 2007. 4.2.4, A.9

[141] S. Dadras and H. R. Momeni, "A novel three-dimensional autonomous chaotic system generating two, three and four-scroll attractors," *Physics Letters A*, vol. 373, no. 40, pp. 3637–3642, 2009. 4.4.2, 4.4.2, A.10

[142] J. C. Sprott, "A dynamical system with a strange attractor and invariant tori," *Physics Letters A*, vol. 378, no. 20, pp. 1361–1363, Apr. 2014. 4.4.3

[143] S. Chatterjee and J. S. Simonoff, *Handbook of Regression Analysis*. Hoboken, New Jersey: Wiley, 2013. 5.2.2

[144] Y. Benjamini and D. Yekutieli, "False Discovery Rate–Adjusted Multiple Confidence Intervals for Selected Parameters," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 71–81, Mar. 2005. 6.2

APPENDIX A

Equations of Dynamical Systems

A.1 Two-Dimensional Damped Oscillator with Linear Dynamics

As discussed in Section 3.4.1, the governing equations for the two-dimensional damped oscillator with linear dynamics are represented by

$$\begin{aligned} \dot{x}_1 &= -0.1x_1 + 2x_2, \\ \dot{x}_2 &= -2x_1 - 0.1x_2. \end{aligned} \tag{A.1.1}$$

A.2 Three-Dimensional Linear System

The equations for the three-dimensional linear system are

$$\begin{aligned} \dot{x}_1 &= -0.1x_1 + 2x_2, \\ \dot{x}_2 &= -2x_1 - 0.1x_2, \\ \dot{x}_3 &= -0.3x_3, \end{aligned} \tag{A.2.2}$$

as discussed in Section 3.4.1.

A.3 Two-Dimensional Damped Oscillator with Cubic Dynamics

As mentioned in Section 3.4.2, the equations for the cubic system are given by

$$\begin{aligned} \dot{x}_1 &= -0.1x_1^3 + 2x_2^3, \\ \dot{x}_2 &= -2x_1^3 - 0.1x_2^3. \end{aligned} \tag{A.3.3}$$

A.4 Lotka-Volterra System

The predator-prey equations are represented as

$$\begin{aligned} \dot{x}_1 &= \alpha x_1 - \zeta x_1 x_2, \\ \dot{x}_2 &= \delta x_1 x_2 - \gamma x_2, \end{aligned} \tag{A.4.4}$$

where the prey birth rate $\alpha = 1$ and the predator death rate $\delta = -1$, and the interaction parameters $\zeta = -1$ and $\gamma = 1$ [136]. See Section 3.4.2.

A.5 Rossler System

In Section 3.4.2, the Rossler system is provided in detail as

$$\begin{aligned} \dot{x}_1 &= -x_2 - x_3, \\ \dot{x}_2 &= x_1 + ax_2, \\ \dot{x}_3 &= b + x_3(x_1 - c), \end{aligned} \tag{A.5.5}$$

where $a = 0.2$, $b = 0.2$, and $c = 5.7$ [130].

A.6 Lorenz System

The Lorenz chaotic system is represented by

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2, \\ \dot{x}_3 &= x_1 x_2 - \zeta x_3, \end{aligned} \tag{A.6.6}$$

with the values of the original parameters $\sigma = 10$, $\rho = 28$, and $\zeta = 8/3$ [46]. See Section 3.4.2 for more details.

A.7 Van der Pol oscillator

Section 3.4.3 offers further information on the Van der Pol oscillator, represented as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \mu(1 - x_1^2)x_2 - x_1, \end{aligned} \tag{A.7.7}$$

where $\mu = 1.2$ controls the nonlinear damping level of the system [64].

A.8 Duffing oscillator

The Duffing oscillator, described in Section 3.4.3, provides an alternative cubic nonlinear system that can represent chaos and often models a spring-damper-mass system that contains a spring with a restoring force of $f(\zeta) = -\kappa\zeta - \epsilon\zeta^3$, where $\epsilon > 0$ represents a hard spring [64]. However, when $\epsilon < 0$, it represents a soft spring and is given by:

$$\ddot{\zeta}_1 + \gamma\dot{\zeta} + (\kappa + \epsilon\zeta^2)\zeta = 0. \tag{A.8.8}$$

Converting $x = \zeta$ and $y = \dot{\zeta}$ transforms Eq. (A.8.8) to

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\gamma x_2 - \kappa x_1 - \epsilon x_1^3. \end{aligned} \tag{A.8.9}$$

Here, the Duffing oscillator was generated using the parameter values for which the equations do not represent chaotic behaviour: $\kappa = 1$, $\gamma = 1$, and $\epsilon = 5$ [64].

A.9 Thomas System

In Section 4.2.4, the Thomas system is represented by

$$\begin{aligned} \dot{x}_1 &= \sin(x_2) - \zeta x_1, \\ \dot{x}_2 &= \sin(x_3) - \zeta x_2, \\ \dot{x}_3 &= \sin(x_1) - \zeta x_3, \end{aligned} \tag{A.9.10}$$

where $\zeta = 0.208186$ [140].

A.10 Dadras System

The Dadras equations are described as

$$\begin{aligned} \dot{x}_1 &= x_2 - \alpha x_1 + \zeta x_2 x_3, \\ \dot{x}_2 &= \upsilon x_2 - x_1 x_3 + x_3, \\ \dot{x}_3 &= \delta x_1 x_2 - \eta x_3, \end{aligned} \tag{A.10.11}$$

with the values of the original parameters $\alpha = 3$, $\zeta = 2.7$, $\upsilon = 4.7$, $\delta = 2$, and $\eta = 9$ [141]. Section 4.4.2 provides further detail regarding the system.

A.10.1 Sprott System

The following equations describe the Sprott system:

$$\begin{aligned} \dot{x}_1 &= x_2 + 2x_1 x_2 + x_1 x_3, \\ \dot{x}_2 &= 1 - 2x_1^2 + x_2 x_3, \\ \dot{x}_3 &= x_1 - x_1^2 - x_2^2. \end{aligned} \tag{A.10.12}$$

See Section 4.4.3 for more details.

A.11 Nonlinear Pendulum Motion Model

The pendulum motion is described by the angle $x_1(t)$ and the angular velocity $x_2(t)$ over time t [136]:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\alpha x_2 + \zeta \sin(x_1), \end{aligned} \tag{A.11.13}$$

where $\alpha = -0.25$ and $\zeta = -5$ [136]. Section 4.4.4 provides further information regarding the nonlinear pendulum.