A COMPREHESIVE ANALYSIS OF DATA-DRIVEN IDENTIFICATION OF ORDINARY DIFFERENTIAL EQUATIONS

A THESIS

Presented to the Department of Mathematics and Statistics

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computational and Applied Mathematics

Committee Members:

Seungjoon Lee, Ph.D. (Chair)
Paul Sun, Ph.D.
Tangan Gao, Ph.D.

College Designee:

Will Murray, Ph.D.

By Alexis Guevara

B.A., 2023, Occidental College

July 2025

ABSTRACT

This thesis presents a comprehensive framework for data-driven identification of nonlinear dynamical systems, combining sparse regression techniques with dynamical systems theory to recover governing equations directly from observational data. Focusing on the Van der Pol oscillator as a benchmark system, the study develops a hybrid methodology that integrates LASSO regularization for sparse term selection with multiple linear regression (MLR) through ordinary least squares (OLS) refinement for unbiased coefficient estimation. The framework is rigorously validated across multiple scenarios, including baseline, noise-corrupted, and forced system configurations, demonstrating robust performance even with significant measurement noise ($\sigma = 0.1$). Key innovations include the systematic incorporation of Takens' Embedding Theorem for state-space reconstruction from partial observations and the development of adaptive function libraries capable of capturing both intrinsic dynamics and external forcing effects.

The methodology is further applied to real-world electrocardiogram (ECG) signals from the PTB Diagnostic Database, where it successfully reconstructs pathological cardiac dynamics using delay-coordinate embeddings. Results show that embedding dimensions $m = 3$ optimally balance accuracy and complexity for forced Van der Pol systems, while $m = 7$ delays are required to capture essential features of ECG waveforms. The framework maintains clinical relevance by identifying physiologically plausible dynamical patterns, though challenges remain in translating delay-coordinate terms into interpretable models.

Theoretical contributions include observed validation of sparse regression for nonlinear system identification and practical insights into parameter selection for delay embedding. Practical applications span engineering systems and biomedical signal processing, with

observations being made for arrhythmia characterization in clinical settings. Limitations are discussed, including sensitivity to derivative estimation errors and the need for domain-specific adaptations. Future research directions highlight opportunities for automated parameter optimization and hybrid physics-informed function libraries.

By bridging mathematical rigor with practical implementation, this work advances the field of data-driven discovery, providing both a versatile toolkit for system identification and a template for developing interpretable models of complex dynamical systems across scientific disciplines. The results demonstrate that sparse, physically meaningful models can achieve competitive performance while retaining the transparency needed for scientific validation and engineering applications.

ACKNOWLEDGEMENTS

This thesis represents the culmination of my academic journey that would not have been possible without the full support and guidance of my family, closest friends, and those who have shaped my personal and intellectual growth.

First, I wish to express my wholehearted gratitude to my parents, whose sacrifices and resilience have laid the foundation for my achievements. My father, through his strong work ethic and commitment to provide for our family after immigrating from Mexico, has embedded the perseverance needed to pursue ambitious goals. My mother, whose insights and unconditional support have been continuous in my life, has been significant in bringing me to this central moment in my academic career. Their continuing belief in my potential has deeply influenced the person I have become, and for this, I remain eternally thankful.

I am grateful to all my siblings, whose presence has been a source of motivation and encouragement. My older brother, Humberto, has been both a role model and strong supporter throughout my academic pursuits, offering guidance and inspiration during challenging times. My younger sisters, Kenya and Lesley, have brought immeasurable joy and emotional support, always reminding me of the importance of perseverance with their optimism and kindness. Witnessing their growth has been a privilege, and I look forward to their future accomplishments.

To my partner, Maura, I owe a debt of gratitude that words cannot fully capture. Your everlasting encouragement, patience, and love have been a constant source of strength, empowering me to navigate the challenges of this journey with confidence and determination. Your belief in me has been a driving force behind all my work, and for that, I am endlessly grateful.

I extend my deepest appreciation to my thesis advisor, Dr. Seungjoon Lee, whose mentorship has been invaluable to my development as a researcher. Your insightful guidance, patience, and dedication to fostering academic excellence have not only shaped this work but have also inspired me to pursue rigor and precision in all aspects of scholarship. Without your support, this thesis would not have been possible.

I would also like to sincerely thank my committee members, Dr. Paul Sun and Dr. Tangan Gao, for their time, expertise, and constructive feedback throughout this process. Your contributions have strengthened this work and broadened my perspective on the subject.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

The Van der Pol oscillator stands as one of the most fundamental nonlinear dynamical systems in mathematical physics, first formulated by Dutch engineer Balthasar van der Pol in 1920 to model stable oscillation circuits in early vacuum tube technology. Defined by the second-order differential equation $\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0$, which can be decomposed into the first-order system of differential equations $\frac{dx}{dt} = v,\ \frac{dv}{dt} = \mu(1 - x^2)v - x$, this oscillator exhibits several mathematically rich phenomena that distinguish it from linear harmonic systems. Most notably, for positive damping parameters $\mu > 0$, the system exhibits limit cycle behavior, where trajectories converge to a stable periodic orbit regardless of initial conditions. This characteristic makes it invaluable for modeling self-sustained oscillations in electrical circuits, neuronal activity, and cardiac rhythms. However, unlike linear oscillators that can be solved exactly using analytical methods, the Van der Pol system's nonlinear damping term $\mu(1 - x^2)\frac{dx}{dt}$ prevents exact analytical solutions in most cases, requiring numerical methods for approximations.

In contemporary applied mathematics, modern challenges arise from systems that generate abundant data but lack theoretical models. Examples include biological rhythms, engineering vibrations, and financial systems. The critical need to extract governing equations directly from observations has led to advances in data-driven identification techniques, which enable prediction, control, and deeper investigation of complex dynamics. The development of sparse regression techniques in dynamical systems, notably promoted by Brunton et al.'s Sparse Identification of Nonlinear Dynamics (SINDy) framework, has revealed new possibilities for

facilitating the discovery of such nonlinear models directly from data. These progressions intersect with Takens' Embedding theorem from dynamical systems theory, which provides a mathematical foundation for reconstructing the full state space of a system from limited observations. The Van der Pol oscillator, with its well-studied nonlinear behavior, serves as an ideal benchmark for testing these system identification methods with the aim of recovering governing equations from observed time series data. These observations can particularly be extended to real-world scenarios involving various forms of perturbations, such as noise and external forcing, and partial observations.

1.2 Motivation

Despite the Van der Pol oscillator's well-established theoretical properties under ideal conditions, significant challenges emerge when attempting to identify its dynamics from real-world data. Such data are often corrupted by noise, subject to external forcing, or limited to partial measurements of the system's state. Traditional analytical approaches, such as perturbation methods or averaging techniques, frequently fail to provide accurate models under these non-ideal conditions. This limitation is particularly pronounced in modern applications, including biomedical engineering, where the oscillator models cardiac pacemaker cells, and aerospace systems, where it approximates structural vibration modes.

Three key gaps in current knowledge motivate this research. First, while sparse regression methods such as Multiple Linear Regression (MLR) and the Least Absolute Shrinkage and Selection Operator (LASSO) have shown promise for system identification, their comparative performance in handling different types of perturbations, such as additive noise, linear forcing, and periodic forcing, has not been comprehensively assessed. Second, the effects of numerical differentiation errors on performance require further investigation, as derivative

estimation is highly sensitive to noise and can significantly degrade model accuracy. Third, the potential of combining Takens' Embedding theorem with modern regression techniques for systems with partial observations has not been fully explored, despite its theoretical promise for reconstructing hidden dynamics. This thesis addresses these gaps by developing a robust framework that integrates sparse regression with dynamical systems theory, ensuring accurate identification of governing equations even under non-ideal conditions.

1.3 Objectives

The primary objective of this thesis is to develop and validate a comprehensive framework for data-driven identification of dynamical systems. The research establishes four key goals to advance the field of nonlinear system identification. The first objective focuses on creating a robust workflow for recovering the Van der Pol equations from synthetic data. This involves implementing numerical solution methods such as the fifth-order Runge-Kutta algorithm, constructing function libraries that span a wide range of potential dynamical terms, and systematically comparing the performance of MLR and LASSO regression methods through metrics such as coefficient error and term selection accuracy. The second objective extends this analysis to non-ideal conditions by introducing three classes of perturbation: additive Gaussian noise, linear forcing, and periodic forcing. For each case, the study quantifies the robustness of identification methods through phase-space reconstruction quality and prediction accuracy. The third objective investigates strategies to enhance identifiability, including the use of time-delay embedding guided by Takens' theorem for systems with partial observations and critical selection of the LASSO penalty parameter to optimize model sparsity. The final objective bridges theory with application by testing the framework on real-world data, specifically electrocardiogram (ECG) signals from the PTB Diagnostic Database. This practical validation

assesses the method's ability to capture clinically relevant features of cardiac rhythms while handling noise and measurement constraints.

The broader impact of this work lies in its balanced approach to model identification, emphasizing both simplicity and accuracy, even when faced with real-world challenges such as noise corruption and incomplete measurements. By integrating theoretical insights with computational tools, the framework provides a versatile toolkit for discovering interpretable dynamical models from data.

1.4 Thesis Structure

This thesis is structured to observe the theoretical foundations that will progressively develop the methodological framework, validate it under diverse conditions, and demonstrate its practical utility. Chapter 2 provides a comprehensive literature review covering the Sparse Identification of Nonlinear Dynamics (SINDy) framework, which entails its mathematical formulation, algorithmic implementation, and transformative impact on data-driven system identification, with particular attention to its success in recovering dynamical models from data. An examination will be made on the fundamental mathematical properties of the Van der Pol oscillator, including its characteristic limit cycle behavior and analytical challenges posed by its nonlinear damping term. A rigorous comparative analysis of regression methods considered and their respective benefits and drawbacks in term selection and noise resilience for dynamical systems will be conducted. A review of Takens' Embedding Theorem and its role in state-space reconstruction from partial observations and highlighting its cooperation with sparse regression in contemporary applications will conclude the chapter.

Chapter 3 establishes the methodological framework, beginning with numerical solution of the oscillator equations using fifth-order Runge-Kutta integration, followed by data

preprocessing techniques for derivative calculation, and culminating in the implementation and optimization hybrid sparse regression combining LASSO and ordinary least squares (OLS) through MLR. Chapters 4 presents the core analytical results, evaluating the framework's performance on the baseline Van der Pol system, its noisy variants, and forced systems with linear or periodic perturbations, systematically evaluating coefficient recovery accuracy, term selection performance, and state-space reconstruction across each case. Chapter 4 introduces quantitative metrics and qualitative assessments for identification robustness under such effects. The application of Taken's Embedding Theorem for the dynamical system will also be explored. Chapter 4 maintains a consistent analytical structure, including simulation setup, regression implementation, function reselection, and comprehensive results validation through both sparse equations and graphical representations.

Chapter 5 transitions to real-world applications, focusing specifically on biological systems with detailed examination of electrocardiography (ECG) signals. This chapter outlines the necessary adaptations of the methodology for physiological data, including preprocessing steps and parameter selection, and interprets the results in a clinical context. Chapter 6 synthesizes the findings, discussing the framework's strengths and limitations, and concludes with concrete directions for future research. Finally, Chapter 7 summarizes the key contributions and their implications for data-driven discovery in nonlinear dynamics.

This structured approach ensures a logical progression from theoretical foundations to empirical validation, maintaining rigor while addressing practical challenges in system identification. The integration of synthetic and real-world case studies underscores the framework's versatility and potential for broader scientific and engineering applications.

CHAPTER 2

LITERATURE REVIEW

2.1 Sparse Identification of Nonlinear Dynamics

The Sparse Identification of Nonlinear Dynamics (SINDy) framework represents a transformative advancement in data-driven system identification, offering a rigorous methodology for discovering governing equations directly from time-series data. Brunton, Proctor, and Kutz (2016) introduced SINDys as a method of combining sparse regression techniques with dynamical systems theory to identify simple nonlinear models.

At its core, the method solves an optimization problem that balances accuracy and sparsity, mathematically formulated as:

$$\min_{\Xi} \|X - \Theta(\dot{X})\Xi\|_2{}^2 + \lambda\|\Xi\|_1,$$

where $\dot{X}$ represents the time derivatives of state variables, $\Theta(X)$ is a library of candidate functions, and Ξ contains the sparse coefficients determining active terms in the dynamics, as demonstrated by Corbetta (2020). The L_1-regularization term $\lambda\|\Xi\|_1$ is central to SINDy's success, as it promotes sparsity by driving irrelevant terms to zero, effectively selecting the simplest model that explains the observed data.

This approach is particularly powerful for systems like the Van der Pol oscillator, where the governing equations are inherently sparse, containing only a few dominant terms despite the potential complexity of the function library. Practical implementation of SINDy requires careful consideration of several factors, including the choice of function library, which must be sufficiently rich to capture the system's dynamics without becoming computationally intractable, and the accurate computation of derivatives from noisy data. Recent extensions of SINDy have expanded its applicability to controlled systems, stochastic dynamics, and partial differential

equations, while theoretical analyses have established conditions under which the true governing equations can be reliably recovered. However, challenges persist, particularly in high-noise dynamical conditions or when dealing with partial observations, where the integration of techniques like delay-coordinate embedding becomes essential.

2.2 Ordinary Differential Equations

Ordinary differential equations (ODEs) form the mathematical foundation for modeling continuous dynamical systems across physics, engineering, and biology. An ODE is defined as an equation involving a single independent variable, typically time t, and one or more derivatives of a dependent variable x with respect to that variable. The general form of an n-th order ODE is:

$$F\left(t, x, x', x'', \ldots, x^{(n)}\right) = 0,$$

where $x^{(k)}$, for $k \in \mathbb{Z}^+$, denotes the k-th derivative of x. The order of an ODE corresponds to its highest derivative. For example, the Van der Pol equation:

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

is second-order due to the presence of $\frac{d^2x}{dt^2}$. Initial conditions, specified as values of x and its derivatives at a particular time, are essential for unique solutions to exist.

A critical classification of ODEs distinguishes between linear and nonlinear forms. Linear differential equations are linear in the unknown function and its variables and maintain the form $a_0(t)x + a_1(t)x' + a_2(t)x'' + \cdots + a_n(t)x^{(n)} = b(t)$, where $a_0(t), \ldots, a_n(t)$ and $b(t)$ are arbitrary differential functions that do not need to be linear, and $x', \ldots, x^{(n)}$ are the successive derivatives of an unknown function x of the variable t. These equations comply with the superposition principle, allowing solutions to be constructed from homogeneous and particular

components. In contract, nonlinear ODEs violate this principle due to nonlinear terms, introducing phenomena such as limit cycles and bifurcation. The distinction has profound implications for solution methods. While linear ODEs often allow exact analytical solutions through techniques like characteristic equations or Laplace transforms, nonlinear ODEs typically require numerical approaches or approximation schemes.

The study of ODEs extends beyond solution techniques to encompass qualitative analysis of long-term behavior through phase portraits, stability analysis, and bifurcation theory. These tools are essential for understanding dynamical systems where explicit solutions may be unattainable but dynamical features like limit cycles emerge from the interactions of nonlinearity and noisy conditions. The transition from linear to nonlinear ODEs marks a fundamental change in modeling complexity, where data-driven methods like SINDy link theoretical complexity and observational study.

2.3 The Van der Pol Oscillator

The Van der Pol oscillator, first introduced in 1920 by Balthasar van der Pol, is a cornerstone of nonlinear dynamical systems, renowned for its rich mathematical behavior and broad applicability that has profoundly influenced the study of oscillatory phenomena across multiple disciplines. The oscillator is governed by the second-order differential equation:

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

which can be rewritten as a first-order system:

$$\frac{dx}{dt} = v \quad , \quad \frac{dv}{dt} = \mu(1 - x^2)v - x.$$

Here, t represents time, x denotes position, v indicates velocity, and μ serves as a scalar parameter that determines both the nonlinearity and damping strength. Through μ, the system

displays a spectrum of distinctive nonlinear damping behaviors. For $\mu > 0$, the system exhibits limit cycle behavior, where trajectories converge to a stable periodic orbit regardless of initial conditions, distinguishing it from linear harmonic oscillators.

As investigated by Kovacic and Mickens (2012), the nonlinear damping term $\mu(1 - x^2)\frac{dx}{dt}$ introduces amplitude-dependent energy dissipation, where when $|x| > 1$, the damping is positive and energy is dissipated, while for $|x| < 1$, the damping is negative and energy is injected, leading to self-sustained oscillations. The absence of exact analytical solutions for $\mu > 0$ necessitates numerical methods for studying its behavior, with techniques like Runge-Kutta integration providing reliable approximations. Beyond its theoretical significance, the Van der Pol oscillator has found a multitude of applications across diverse fields. Beyond its theoretical significance, the Van der Pol oscillator has found widespread applications across diverse fields. The oscillator's versatility and well-characterized behavior make it an ideal benchmark system for evaluating data-driven identification methods, specifically in the presence of perturbations like noise and external forcing.

The Van der Pol oscillator's mathematical depth extends to its bifurcation behavior, specifically the Hopf bifurcation at $\mu = 0$, where the system transitions from a stable fixed point to a stable limit cycle. This bifurcation has been extensively studied in nonlinear dynamics, offering insights into the emergence of periodic behavior in real-world systems. Furthermore, forced variations of the oscillator exhibit complex phenomena, making them valuable systems for studying phase synchronization real-world systems.

Applications of the Van der Pol oscillator span numerous fields. In electrical engineering, it models oscillations in nonlinear electronic circuits and systems. In biology, it supports a simplified representation of neuronal action potentials that has been used to study cardiac

rhythms and electrocardiogram (ECG) waveforms. In mechanical and aerospace engineering, the oscillator describes self-excited vibrations in structures.

To illustrate the oscillator's behavior, consider the series of phase portraits, which depict the limit cycle for $\mu = 0.5, 1, 2, 3$ in Figure 1.

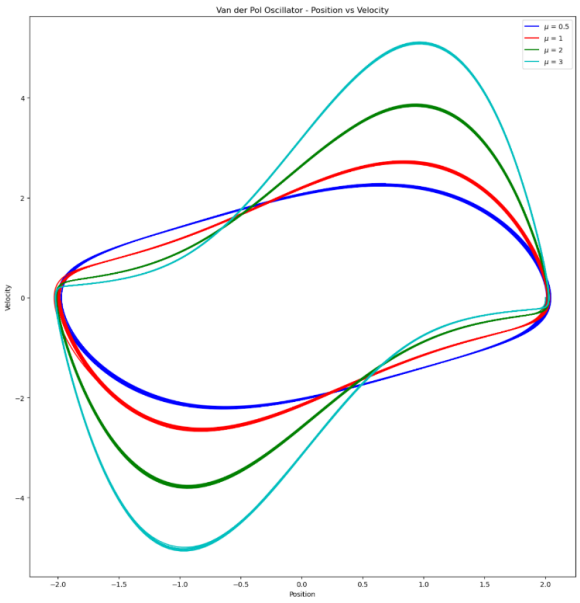


FIGURE 1. Phase portrait of Van der Pol oscillator for $\mu = 0.5, 1, 2, 3$.

This figure highlights the characteristic limit cycle and the influence of initial conditions, reinforcing the oscillator's role as a standard example of nonlinear dynamics. The Van der Pol oscillator's combination of theoretical depth and practical relevance ensures its continued importance in both fundamental research and applied sciences, serving as a link between abstract mathematical concepts and real-world dynamical systems.

2.4 Regression Methods for Dynamical Systems Identification

2.4.1 Multiple Linear Regression (MLR)

Multiple Linear Regression (MLR) is a foundational statistical method for modeling the relationship between a dependent variable and multiple independent variables. Mathematically, MLR solves the ordinary least squares (OLS) problem:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$$

where $\boldsymbol{y}$ is the vector of observed outcomes, $\boldsymbol{X}$ is the design matrix of candidate functions, and $\boldsymbol{\beta}$ contains the coefficients to be estimated. Brunton, Proctor, and Kutz (2016) propose the utilization of MLR through OLS in dynamical systems identification is to provide unbiased estimates of the governing equations when the true model structure is known and the function library is well-specified. A key advantage of MLR is its computational efficiency and interpretability, as the solution can be obtained analytically via the normal equations $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\,\boldsymbol{y}$. However, MLR suffers from significant limitations in high-dimensional settings where the function library may contain a multitude of candidate terms. When predictors are correlated or the number of terms exceeds the number of observations, MLR tends to overfit noise and retain irrelevant terms, leading to poor generalization. In the context of identifying the Van der Pol oscillator, MLR serves as a baseline method to highlight the advantages of more advanced techniques like LASSO, particularly in noisy dynamical conditions where its performance degrades markedly.

2.4.2 Least Absolute Shrinkage and Selection Operator (LASSO)

The Least Absolute Shrinkage and Selection Operator (LASSO) extends OLS by incorporating an ℓ_1-norm penalty to promote sparsity:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

Kiser, Guskov, Rébillat, and Ranc (2023) examine the ℓ_1-penalty driving the coefficients of irrelevant terms to exactly zero, effectively performing variable selection while fitting the model. This property is invaluable for identifying the Van der Pol oscillator's dynamics, where the true governing equations contain only a few nonzero terms $(e.g., x, v, x^2v)$ among a large function library. LASSO's advantages include inherent noise robustness, as the penalty acts as a filter, and

interpretability since the sparse models reveal dominant physical features. However, its performance depends critically on the regression parameter λ, selected here through structured data splitting. Limitations include potential overshrinkage of large coefficients and sensitivity to highly correlated predictors, which were reduced through OLS refitting and iterative reselection. Therefore, LASSO has a larger likelihood to be superior to MLR for the Van der Pol identification when considering dynamical systems under perturbations in preserving key dynamical features.

2.5 Takens' Embedding Theorem and State-Space Reconstruction

Yap and Rozell (2011) investigate Takens' Embedding Theorem as providing a framework for reconstructing the full state-space dynamics of a system from limited observations, a capability that is critical for data-driven identification in real-world applications. The theorem states that for an observable f and time delay τ, the delay-coordinate map:

$$\Phi(x) = \left(f(x), f\left(\phi_{\{-\tau\}}(x)\right), \ldots, f\left(\phi_{\{-(2m+1)\tau\}}(x)\right)\right),$$

embeds a m-dimensional attractor into $\mathbb{R}^{2m+1}$, preserving its topological and dynamic properties. This result enables the reconstruction of the system's phase space from a single time series, circumventing the need for direct measurement of all state variables. Practical implementation of Takens' theorem involves two critical steps: selecting an appropriate time delay τ and determining the embedding dimension m. The time delay is often chosen using mutual information, which balances the trade-off between redundancy and independence in the delay coordinates, while the embedding dimension is typically estimated using the topological unfolding algorithm, ensuring that the attractor is fully unfolded without excessive dimensionality. In the context of system identification, Takens' embedding integrates effectively with sparse regression techniques like SINDy, allowing for the discovery of governing equations

even when only partial observations are available. For example, in the Van der Pol oscillator, the velocity variable $v(t)$ can be reconstructed from measurements of $x(t)$ alone, enabling complete system identification from limited data. Moreover, the delay-coordinate representation enhances robustness to noise by effectively distributing measurement errors across multiple dimensions. This combined approach has proven particularly effective in challenging scenarios, such as systems subject to high noise levels or external forcing and has broad applicability in fields ranging from physiology to engineering.

CHAPTER 3

METHODOLOGY

The methodology developed in this thesis provides a systematic framework for data-driven identification of the Van der Pol oscillator and its variants, integrating numerical techniques, sparse regression, and dynamical systems theory. The approach consists of six key components: data generation, derivative calculation, function library construction, regression techniques, function reselection, and validation metrics, each designed to address specific challenges in system identification while maintaining robustness to noise and model complexity.

3.1 Data Generation

The synthetic data generation process for this study was designed to systematically explore the behavior of the Van der Pol oscillator under various dynamical conditions, including the baseline system, noise-corrupted dynamics, and forced cases. The oscillator's governing equations, expressed as the first-order system of ordinary differential equations $\frac{dx}{dt} = v$ and $\frac{dv}{dt} = \mu(1 - x^2)v - x$, were numerically approximated using a fifth-order Runge-Kutta method (RK45) with a fixed time step of $\Delta t = 0.01$ over the interval $t \in [0, 50]$. This time step was selected to ensure sufficient resolution of the system's dynamics while maintaining computational efficiency. The RK45 method was chosen for its balance between accuracy, with a local truncation error of $O(\Delta t^5)$, and stability for stiff systems.

The parameter $\mu = 2$ was selected to produce pronounced nonlinear limit cycle behavior, with initial conditions set to $x(0) = 2$ and $v(0) = 0$ to capture the system's stable oscillations. To investigate robustness under noise, additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ was introduced to the state variables, with noise levels $\sigma^2 \in \{0, 0.1\}$ representing scenarios from ideal to

experimentally realistic conditions. The amplitude $\sigma = 0.1$ was chosen to reflect typical measurement noise encountered in real-world settings.

For the forced oscillator variants, two distinct cases were considered: a linearly forced system with an additional term $C \cdot t$ in the velocity equation or a periodically forced system incorporating $D \cdot \sin(t)$ in the velocity equation. The forcing coefficients $C \in R$ and $D \in \mathbb{R}$ were carefully tuned to ensure perturbations were noticeable but did not dominate the intrinsic dynamics of the Van der Pol oscillator. The resulting trajectories were sampled at uniform intervals, producing time-series data for $x(t)$, $v(t)$, and their derivatives, which were stored for subsequent analysis. This comprehensive data generation strategy enabled systematic evaluation of the identification framework across different dynamical systems.

3.2 Derivative Calculation and Numerical Integration

Derivative estimation from the generated data was performed using a forward Euler difference scheme, chosen for its computational simplicity and ease of implementation. The first derivatives were approximated as:

$$\frac{dx}{dt} \approx \frac{x_{\{i+1\}} - x_i}{\Delta t} \quad \text{and} \quad \frac{dv}{dt} \approx \frac{v_{\{i+1\}} - v_i}{\Delta t},$$

where Δt is the fixed time step between samples. While Euler's method introduces a local truncation error of $O(\Delta t)$, its computational efficiency made it suitable for initial investigations, particularly when studying the effects of noise on derivative estimation. Nevertheless, the method's vulnerability to numerical instability and noise amplification warrants discussion.

For numerical integration, the explicit recursive formulas presented as:

$$x_{\{i+1\}} \approx x_i + \Delta t \cdot \frac{dx}{dt} \quad \text{and} \quad v_{\{i+1\}} \approx v_i + \Delta t \cdot \frac{dv}{dt}$$

were employed to reconstruct state trajectories from the estimated derivatives. The approach maintained consistency with the forward Euler scheme used for differentiation, though it inherits the same $O(\Delta t)$ error scaling. In systems with forcing terms, the error bounds expanded to $\mathcal{O}(\Delta t \cdot \max(|C|, |D|))$, where C and D represent the magnitudes of linear and periodic forcing respectively. The derivative computation framework provides an essential link between the continuous-time system dynamics and the discrete-time regression problem, with carefully characterized errors that can support the interpretation of identification results when observing results.

The choice of finite difference methods over more sophisticated approaches was deliberate, allowing isolation of the sparse regression performance from potential effects introduced by advanced differentiation techniques. This design decision facilitated clearer interpretation of results regarding the framework's noise robustness, as any degradation in performance could be directly attributed to the regression methodology rather than preprocessing steps.

3.3 Function Library Construction

The function library was constructed to encompass a broad range of potential dynamical terms while remaining computationally controllable. Polynomial terms up to third order formed the core of the library, including $x, v, x^2, x \cdot v, v^2, x^3, x^2 \cdot v, x \cdot v^2$ and v^3. These terms were selected to capture the canonical nonlinearities expected in the Van der Pol system, particularly the essential $x^2 v$ cross-term representing a nonlinear damping component in $\mu(1 - x^2)v$.

Time-dependent terms, such as t or t^2, were intentionally omitted to enforce time-invariance in the identified models, reflecting the autonomous nature of the baseline Van der Pol system. The choice assumes stationarity in the underlying dynamics, which holds for the baseline

system but may require additional terms in the function library for forced cases. Although the forced systems introduce time-dependent terms, the library remained unchanged due to wanting to observe the impact of the time-dependency on the dynamical system and the adaptation of the model prediction within the set time interval.

For systems with partial observations, the library was expanded using Takens' Embedding Theorem, which enables the reconstruction of hidden dynamics from time-delayed observations of a single variable. When only $x(t)$ was measurable, delay coordinates $x(t - \tau), x(t - 2\tau), \ldots, x(t - m\tau)$ were computed, with $\tau \in \mathbb{Z}^+$ selected through mutual information analysis and $m \in \mathbb{Z}^+$ chosen to satisfy the embedding dimension criterion $m > 2d + 1$ for the oscillator's two-dimensional state space. These delay terms were combined with the original polynomial library, creating a comprehensive set of candidate functions capable of representing both the system dynamics and any emergent behaviors from the additional forcing effect.

3.4 Regression Techniques

System identification was performed using a two-stage regression approach combining LASSO regression and ordinary least squares (OLS) refinement. The LASSO regression was applied first with a carefully selected penalty parameter $\lambda > 0$ to induce sparsity in the coefficient vector. The optimization problem was solved using coordinate descent, which is particularly efficient for high-dimensional problems with $L1$ regularization. The regularization strength λ was chosen through preliminary testing on synthetic data to balance model complexity and accuracy, with smaller values of λ retaining more terms at the risk of overfitting.

Following the LASSO step, the selected features were then refitted using OLS to obtain unbiased coefficient estimates, moderating the known tendency of LASSO to overshrink large

coefficients. This hybrid approach leveraged LASSO's strength in high-dimensional variable selection while avoiding its limitations through subsequent OLS refinement.

Goyal, Pawan, and Benner (2022) emphasize the critical step of enforcing a threshold scheme in the procedure of constructing sparse equations and plots to remove terms with critically small coefficients that are close to zero. Since the OLS refinement step did not remove any terms considered insignificant, a coefficient threshold of 1×10^{-6} was applied, removing all terms with corresponding coefficients less than the threshold. The regression process was conducted separately for each state derivative $\frac{dx}{dt}$ and $\frac{dv}{dt}$, allowing discovery of distinct dynamical relationships for position and velocity. The entire pipeline was implemented in Python using scikit-learn for the regression components, with custom code for feature library construction and performance evaluation.

3.5 Function Reselection and Model Validation

The identified models underwent rigorous validation through both quantitative and qualitative assessments. Quantitative evaluation compared the predicted derivatives $\frac{dx}{dt}_{pred}$ and $\frac{dv}{dt}_{pred}$ against the true values using time-series plots and phase portrait. Particular attention was paid to the preservation of dynamical features like limit cycle shape and periodicity in the phase portraits.

Qualitative assessment involved direct inspection of the sparse equations to verify recovery of key physical terms (e.g., the nonlinear damping $\mu(1 - x^2)v$) while identifying any extraneous additions. For forced systems, special emphasis was placed on the identification of time-invariant terms representing the time-dependent forcing terms $C \cdot t$ and $D \cdot sin(t)$. A critical validation step involved numerical integration of the discovered models using the same

Δt as the original data generation, comparing the resulting trajectories to the true system behavior in both time and phase space.

The validation process included an iterative function reselection step, where models were refined based on their predictive performance. Terms with coefficients below a specified threshold (typically 1×10^{-6}) were discarded, and the remaining terms were refit using OLS. The final models were evaluated on completely independent test data to assess their generalization capability, completing the end-to-end system identification workflow.

CHAPTER 4

RESULTS

The results presented in this chapter demonstrate the effectiveness of the proposed methodology through comprehensive numerical experiments of the Van der Pol oscillator and its forced variants. The analysis systematically evaluates three cases: the baseline oscillator, linearly forced system, and periodically forced system, each examined under both noiseless and noisy conditions. Key aspects of the investigation include the accuracy of sparse equation recovery, the comparative performance of LASSO and OLS-refined models, and the quantitative assessment of state-space reconstruction through time-domain and phase-space analyses. For forced systems, particular attention is given to the role of Takens' embedding theorem in improving dynamical identification across varying delay dimensions ($m = 0, 1, 2, 3$). The results are organized to first establish baseline performance before examining increasingly complex scenarios, with all findings supported by both analytical comparisons of identified equations and graphical representations of system trajectories.

4.1 Baseline Van der Pol System

The baseline Van der Pol system, governed by the first-order equations

$$\frac{dx}{dt} = v \quad \text{and} \quad \frac{dv}{dt} = \mu(1 - x^2)v - x,$$

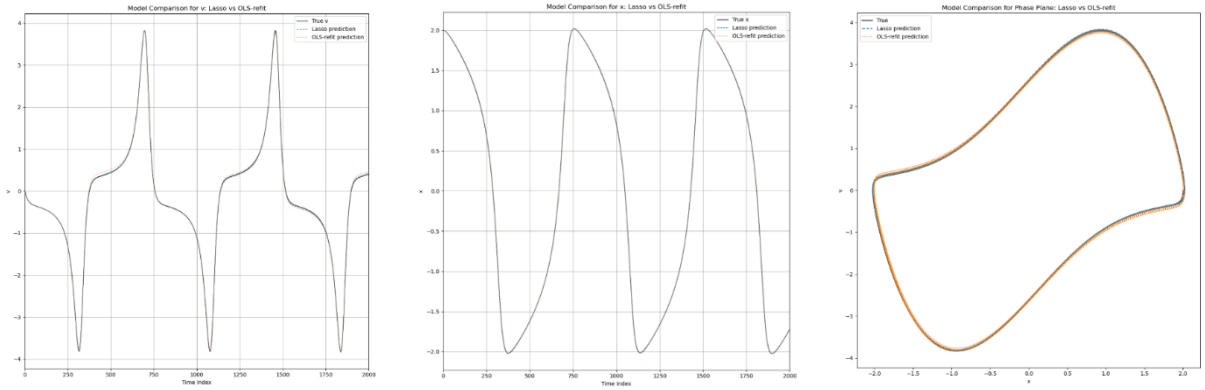
with $\mu = 2$ and initial conditions $x(0) = 2.0$ and $v(0) = 0.0$, served as the foundational test case for evaluating the identification framework. Further investigations of the baseline Van der Pol system governed by the second-order equation

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

with $\mu = 2$ and initial condition $x(0) = 2.0$, is demonstrated in Appendix A. Under noiseless conditions, as illustrated in Figure 2, the hybrid LASSO-OLS regression method successfully recovered all key terms in the governing equations and corresponding coefficients with high accuracy. The identified coefficients for the nonlinear damping term $\mu(1 - x^2)v$ matched the true value within a small margin of error, while terms deemed irrelevant were correctly eliminated during the OLS refinement stage. The sparse regression reduced the initial candidates from 5 to 2 for the equation approximating $\frac{dx}{dt}$ and from 7 to 3 for the equation approximating $\frac{dv}{dt}$ as shown in Figure 4.

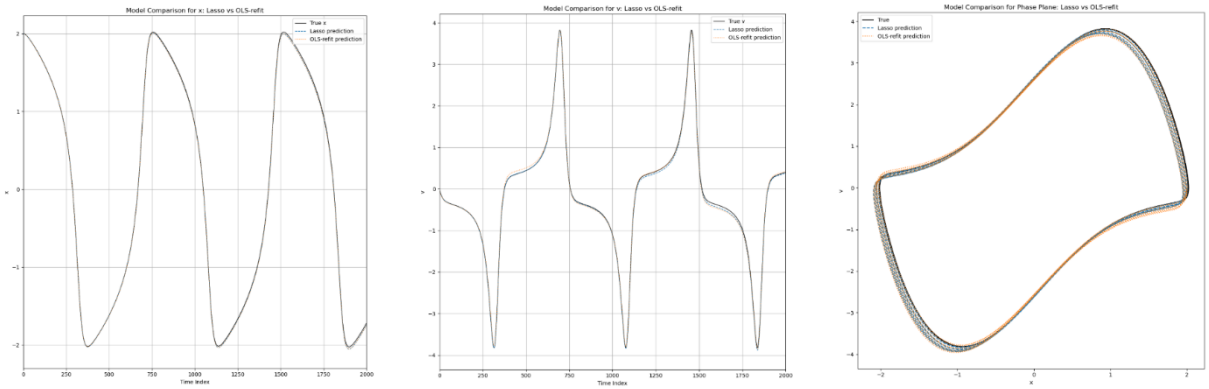
When additive Gaussian noise $(\sigma = 0.1)$ was introduced, the framework remained robust. LASSO alone selected six additional terms with coefficients below 0.01 for modeling $\frac{dx}{dt}$ and four additional terms with coefficients below 0.05 for modeling $\frac{dv}{dt}$, reflecting the expected degradation in performance under noise, as demonstrated in Figure 5. However, the subsequent OLS refinement eliminated most irrelevant terms, yielding a final model structure nearly identical to the noiseless case, displayed in Figure 3. The coefficient for the critical x^2v term exhibited a slight deviation from the true value, while the noise level resulted in a slight change in coefficients for the remaining relevant terms.

State predictions generated by numerically integrating the identified models showed excellent consistency and accuracy with the true system behavior. In both the noiseless and noisy case, both position and velocity trajectories overlapped nearly perfect with the ground truth over the full 50-second simulation. For the noisy system, phase portraits revealed minimal distortion of the limit cycle, with the deviations occurring during increasing changes in velocity. These results confirm the framework's ability to extract accurate dynamical models even when derivative estimates are corrupted by moderate noise.



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 2. Baseline Van der Pol System (Noiseless).



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 3. Baseline Van der Pol System (Noisy).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.996138)
('x_t_0*x_t_0*x_t_0', -0.000843)
('x_t_0*x_t_0*v_t_0', -0.001630)
('x_t_0*v_t_0*v_t_0', -0.005732)
('v_t_0*v_t_0*v_t_0', 0.001824)

# === OLS-refitted coefficients ===
('v_t_0', 1.010024)
('x_t_0*x_t_0*v_t_0', -0.010083)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.944451)
('v_t_0', 1.990389)
('x_t_0*v_t_0', -0.000171)
('x_t_0*x_t_0*x_t_0', -0.007284)
('x_t_0*x_t_0*v_t_0', -1.958206)
('x_t_0*v_t_0*v_t_0', -0.041506)
('v_t_0*v_t_0*v_t_0', 0.002795)

# === OLS-refitted coefficients ===
('x_t_0', -1.002721)
('v_t_0', 1.979598)
('x_t_0*x_t_0*v_t_0', -2.003927)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 4. Feature Selection for Baseline Van der Pol System (Noiseless).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.990944)
('x_t_0*x_t_0', 0.000496)
('v_t_0*v_t_0', -0.000598)
('x_t_0*x_t_0*x_t_0', -0.001125)
('x_t_0*x_t_0*v_t_0', -0.000165)
('x_t_0*v_t_0*v_t_0', -0.006251)
('v_t_0*v_t_0*v_t_0', 0.002340)

# === OLS-refitted coefficients ===
('v_t_0', 1.009636)
('x_t_0*x_t_0*x_t_0', -0.001717)
('x_t_0*x_t_0*v_t_0', -0.009733)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.941703)
('v_t_0', 1.990888)
('v_t_0*v_t_0', 0.000624)
('x_t_0*x_t_0*x_t_0', -0.008070)
('x_t_0*x_t_0*v_t_0', -1.958282)
('x_t_0*v_t_0*v_t_0', -0.042837)
('v_t_0*v_t_0*v_t_0', 0.002926)

# === OLS-refitted coefficients ===
('x_t_0', -1.003459)
('v_t_0', 1.979916)
('x_t_0*x_t_0*v_t_0', -2.005411)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 5. Feature Selection for Baseline Van der Pol System (Noisy).

4.2 Linearly Forced Van der Pol System

The linearly forced system, incorporating an additional time-dependent term $C \cdot t$ with $C = 0.1$ in the velocity equation, has the following form

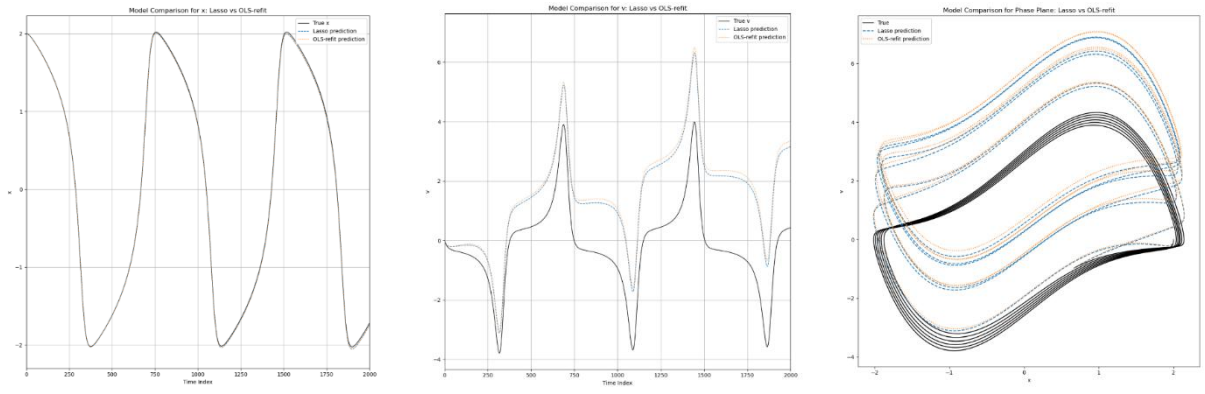
$$\frac{dv}{dt} = \mu(1 - x^2)v - x + C \cdot t,$$

with $\mu = 2$ and initial conditions $x(0) = 2.0$ and $v(0) = 0.0$, presented a more challenging identification scenario. Without delay embedding ($m = 0$), both LASSO and OLS-refined models failed to capture the forcing dynamics, resulting in velocity predictions that diverged systematically from the true solution and insufficient support from other variables in the function library to account for the linear forcing, given in Figure 11. The phase portrait showed a drifting effect as the model's unforced limit cycle failed to account for the accumulating effect of linear forcing, shown in Figure 6.

Application of Takens' embedding with dimensions $m = \{1, 2, 3\}$ substantially improved performance. The OLS-refined models correctly identified the behavior of the true dynamical system over time, as depicted in Figure 7, Figure 8, and Figure 9. The sparse equations presented significant changes than the ground-truth equations since no time-dependent terms were considered in the function library to capture the time-dependent linear forcing term in the

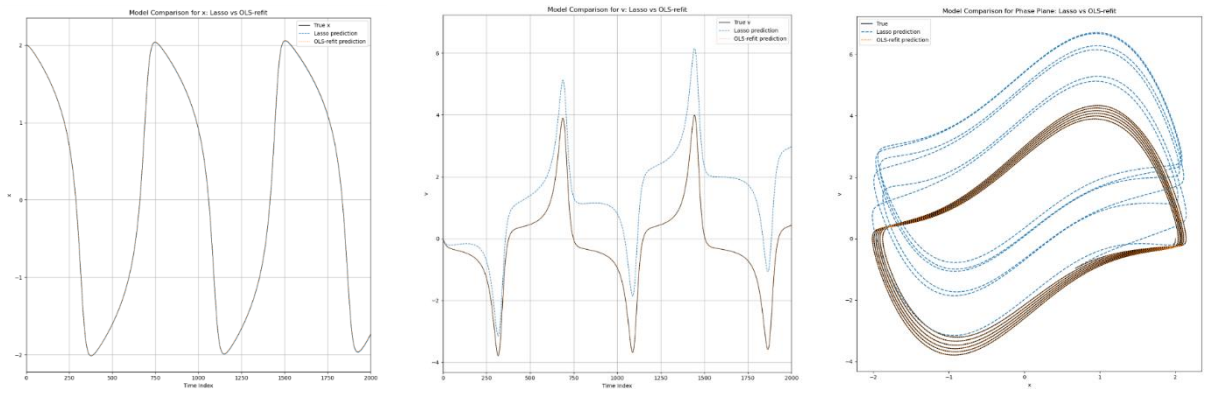
velocity equation, as depicted in Figure 12, Figure 13, and Figure 14. The LASSO models failed to correctly capture the true system dynamics and maintained similar behavior as the models without delay embedding ($m = 0$). Increasing the embedding to $m = 3$ introduced additional delay terms that marginally improved prediction accuracy at the cost of model complexity, which can be observed through the significant increase in the number of terms for the final sparse equations. Since no change was made to the position equation, the predictions remained accurate across all cases. However, as the embedding dimension increased, more terms were presented in the final sparse equations, and the corresponding coefficients obscured the relevancy of the ground-truth variables.

Under noisy conditions ($\sigma = 0.1, m = 3$), the framework maintained reasonable performance, though with increased sensitivity to noise, as plotted in Figure 10. The identified model retained the essential forcing dynamics while showing similar behavior over the 50-second simulation, with the resulting sparse equations shown in Figure 15. The results highlight the challenge of constructing effective sparse equations and models while maintaining long-term predictive accuracy when both noise and forcing are present.



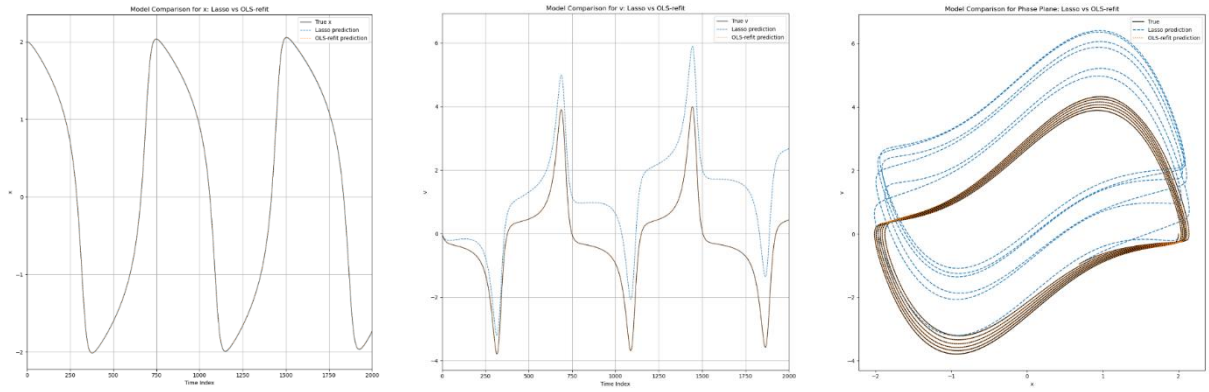
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 6. Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 0$).



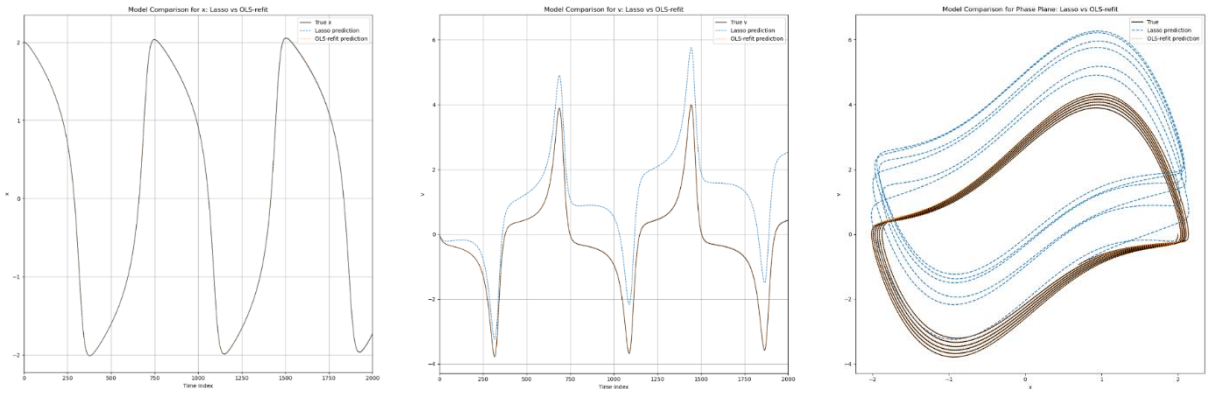
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 7. Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 1$).



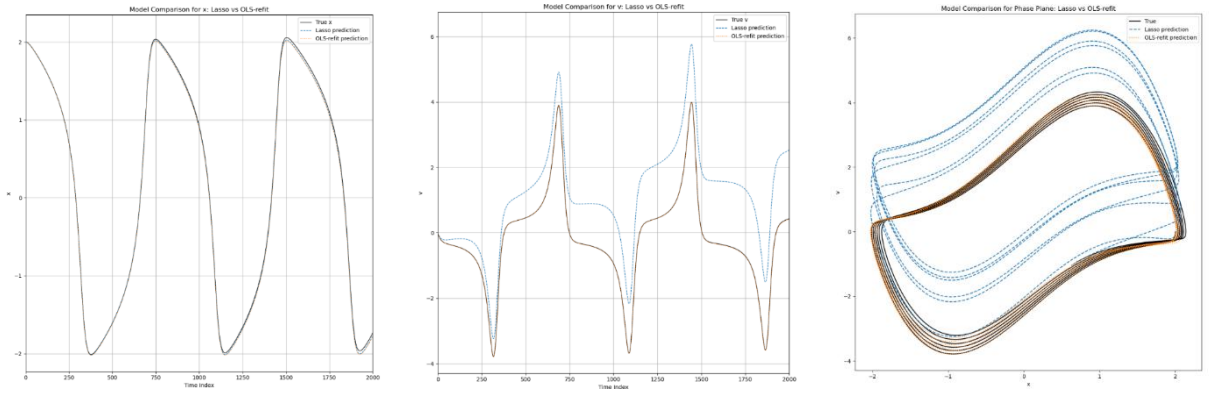
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 8. Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 2$).



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 9. Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 3$).



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 10. Linearly Forced Van der Pol System (Noisy, Embedding Dimension $m = 3$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.998322)
('v_t_0*v_t_0', -0.000352)
('x_t_0*x_t_0*x_t_0', -0.000873)
('x_t_0*x_t_0*v_t_0', -0.002621)
('x_t_0*v_t_0*v_t_0', -0.004894)
('v_t_0*v_t_0*v_t_0', 0.001500)

# === OLS-refitted coefficients ===
('v_t_0', 1.010084)
('x_t_0*x_t_0*x_t_0', -0.001459)
('x_t_0*x_t_0*v_t_0', -0.010071)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.943541)
('v_t_0', 1.899287)
('x_t_0*x_t_0', -0.013183)
('x_t_0*v_t_0', 0.027733)
('v_t_0*v_t_0', -0.014591)
('x_t_0*x_t_0*x_t_0', 0.000375)
('x_t_0*x_t_0*v_t_0', -1.904191)
('x_t_0*v_t_0*v_t_0', -0.079765)
('v_t_0*v_t_0*v_t_0', 0.016416)

# === OLS-refitted coefficients ===
('x_t_0', -0.958778)
('v_t_0', 2.013089)
('x_t_0*x_t_0*v_t_0', -1.967232)
('x_t_0*v_t_0*v_t_0', -0.032162)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 11. Feature Selection for Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 0$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.972336)
('v_t_1', 0.024212)
('v_t_1*v_t_1', -0.000436)
('x_t_0*v_t_0*v_t_1', -0.000045)
('x_t_0*x_t_1*x_t_1', -0.000420)
('x_t_0*v_t_1*v_t_1', -0.001603)
('v_t_0*v_t_0*v_t_0', 0.001740)
('v_t_0*v_t_0*x_t_1', -0.001410)
('v_t_0*x_t_1*x_t_1', -0.001761)
('v_t_0*x_t_1*v_t_1', -0.000747)
('x_t_1*x_t_1*x_t_1', -0.000377)
('x_t_1*v_t_1*v_t_1', -0.001608)

# === OLS-refitted coefficients ===
('v_t_0', 0.750341)
('v_t_1', 0.254714)
('x_t_0*x_t_1*x_t_1', -6.208433)
('x_t_0*v_t_1*v_t_1', -0.000619)
('v_t_0*x_t_1*x_t_1', -0.065123)
('x_t_1*x_t_1*x_t_1', 6.208139)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.910231)
('v_t_0', 0.458391)
('v_t_1', 1.342026)
('x_t_0*x_t_0', -0.008575)
('x_t_0*v_t_1', 0.023336)
('v_t_0*v_t_0', -0.017290)
('x_t_0*x_t_0*v_t_0', -1.287014)
('x_t_0*v_t_0*v_t_0', 0.077277)
('x_t_0*v_t_0*x_t_1', -0.116234)
('x_t_0*v_t_1*v_t_1', -0.081344)
('v_t_0*v_t_0*v_t_0', 0.143069)
('v_t_0*v_t_0*v_t_1', -0.001693)
('v_t_0*x_t_1*x_t_1', -0.453893)
('v_t_0*v_t_1*v_t_1', -0.050984)
('x_t_1*x_t_1*x_t_1', -0.003905)
('x_t_1*v_t_1*v_t_1', -0.123341)
('v_t_1*v_t_1*v_t_1', -0.055654)

# === OLS-refitted coefficients ===
('x_t_0', -0.000000)
('v_t_0', -100.000000)
('v_t_1', 100.000000)
('x_t_0*x_t_0', -0.000000)
('x_t_0*v_t_1', 0.000000)
('v_t_0*v_t_0', -0.000000)
('x_t_0*x_t_0*v_t_0', -0.000000)
('v_t_0*x_t_1*x_t_1', 0.000000)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 12. Feature Selection for Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 1$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.964676)
('v_t_1', 0.015679)
('v_t_2', 0.015349)
('v_t_2*v_t_2', -0.000476)
('x_t_0*x_t_1*x_t_2', -0.000069)
('x_t_0*v_t_1*v_t_2', -0.000581)
('x_t_0*x_t_2*x_t_2', -0.000163)
('x_t_0*v_t_2*v_t_2', -0.001205)
('v_t_0*v_t_0*v_t_0', 0.001537)
('v_t_0*v_t_0*x_t_1', -0.000235)
('v_t_0*x_t_1*v_t_1', -0.000663)
('v_t_0*x_t_1*x_t_2', -0.001064)
('v_t_0*x_t_1*v_t_2', -0.000505)
('v_t_0*x_t_2*x_t_2', -0.000433)
('x_t_1*x_t_1*x_t_2', -0.000196)
('x_t_1*v_t_1*v_t_2', -0.000746)
('x_t_1*x_t_2*x_t_2', -0.000193)
('x_t_1*v_t_2*v_t_2', -0.001181)
('v_t_1*v_t_1*v_t_1', 0.000279)
('x_t_2*x_t_2*x_t_2', -0.000150)
('x_t_2*v_t_2*v_t_2', -0.000444)

# === OLS-refitted coefficients ===
('v_t_0', -0.479156)
('v_t_1', 2.435532)
('v_t_2', -0.956721)
('x_t_0*x_t_2*x_t_2', -47.979464)
('v_t_0*v_t_0*x_t_1', -0.008709)
('v_t_0*x_t_1*x_t_2', -0.303320)
('x_t_1*x_t_1*x_t_2', 297.702522)
('x_t_1*x_t_2*x_t_2', -529.774179)
('x_t_2*x_t_2*x_t_2', 280.051126)
('x_t_2*v_t_2*v_t_2', -0.024807)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.907819)
('v_t_1', 0.462075)
('v_t_2', 1.169114)
('x_t_0*x_t_0', -0.002284)
('x_t_0*v_t_2', 0.014816)
('v_t_0*v_t_0', -0.022867)
('x_t_0*x_t_0*v_t_0', -1.188580)
('x_t_0*v_t_0*v_t_0', 0.217445)
('x_t_0*v_t_0*x_t_1', -0.305564)
('x_t_0*v_t_2*v_t_2', -0.225648)
('v_t_0*v_t_0*v_t_0', 0.155813)
('v_t_0*v_t_0*v_t_2', -0.006081)
('v_t_0*x_t_1*x_t_1', -0.298527)
('v_t_0*x_t_1*x_t_2', -0.015745)
('v_t_0*v_t_1*v_t_2', -0.019471)
('v_t_0*v_t_2*v_t_2', -0.039555)
('x_t_1*x_t_1*x_t_1', -0.001409)
('x_t_1*x_t_1*x_t_2', -0.000079)
('x_t_1*x_t_2*x_t_2', -0.000096)
('x_t_1*v_t_2*v_t_2', -0.188475)
('v_t_1*v_t_2*v_t_2', -0.001016)
('x_t_2*v_t_2*v_t_2', -0.005087)
('v_t_2*v_t_2*v_t_2', -0.029149)

# === OLS-refitted coefficients ===
('v_t_1', -106.280990)
('v_t_2', 106.286872)
('x_t_0*x_t_0', -0.007098)
('x_t_0*v_t_2', -0.010684)
('x_t_0*v_t_0*v_t_0', -0.202555)
('x_t_0*v_t_2*v_t_2', 0.307232)
('v_t_0*x_t_1*x_t_1', 0.015619)
('v_t_0*v_t_1*v_t_2', 9.043946)
('v_t_0*v_t_2*v_t_2', -18.505982)
('v_t_1*v_t_2*v_t_2', 9.635795)
('v_t_2*v_t_2*v_t_2', -0.203330)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 13. Feature Selection for Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 2$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.952747)
('v_t_1', 0.014196)
('v_t_2', 0.013877)
('v_t_3', 0.013602)
('v_t_3*v_t_3', -0.000529)
('x_t_0*v_t_2*v_t_3', -0.000106)
('x_t_0*x_t_3*x_t_3', -0.000127)
('x_t_0*v_t_3*v_t_3', -0.001152)
('v_t_0*v_t_0*v_t_0', 0.001515)
('v_t_0*x_t_1*v_t_2', -0.000570)
('v_t_0*x_t_1*v_t_3', -0.000648)
('v_t_0*x_t_3*x_t_3', -0.000997)
('v_t_0*x_t_3*v_t_3', -0.000015)
('x_t_1*x_t_1*x_t_3', -0.000039)
('x_t_1*v_t_1*v_t_2', -0.000030)
('x_t_1*v_t_1*v_t_3', -0.000253)
('x_t_1*x_t_2*x_t_2', -0.000026)
('x_t_1*x_t_2*x_t_3', -0.000208)
('x_t_1*v_t_2*v_t_3', -0.000931)
('x_t_1*x_t_3*x_t_3', -0.000064)
('x_t_1*v_t_3*v_t_3', -0.001145)
('v_t_1*v_t_1*v_t_1', 0.000421)
('x_t_2*x_t_3*x_t_3', -0.000194)
('x_t_2*v_t_3*v_t_3', -0.000953)
('x_t_3*x_t_3*x_t_3', -0.000054)

# === OLS-refitted coefficients ===
('v_t_0', -0.286112)
('v_t_1', 1.649347)
('v_t_3', -0.363571)
('x_t_0*x_t_3*x_t_3', -49.149392)
('v_t_0*x_t_1*v_t_2', 0.009440)
('v_t_0*x_t_3*x_t_3', -0.297445)
('v_t_0*x_t_3*v_t_3', -0.044954)
('x_t_1*x_t_1*x_t_3', 187.647340)
('x_t_1*x_t_2*x_t_2', -79.938431)
('x_t_1*x_t_3*x_t_3', -225.415244)
('x_t_2*x_t_3*x_t_3', 137.691044)
('x_t_3*x_t_3*x_t_3', 29.164703)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.880914)
('v_t_2', 0.523772)
('v_t_3', 1.029481)
('x_t_0*x_t_0', -0.002613)
('x_t_0*v_t_3', 0.020984)
('v_t_0*v_t_0', -0.026210)
('x_t_0*x_t_0*v_t_0', -1.492306)
('x_t_0*v_t_0*v_t_0', 0.169343)
('x_t_0*v_t_0*x_t_1', -0.232292)
('x_t_0*v_t_3*v_t_3', -0.145318)
('v_t_0*v_t_0*v_t_0', 0.103247)
('v_t_0*x_t_1*x_t_1', -0.223386)
('v_t_0*x_t_1*x_t_2', -0.002501)
('v_t_0*x_t_1*x_t_3', -0.007917)
('v_t_0*x_t_2*x_t_2', -0.007248)
('v_t_0*x_t_2*x_t_3', -0.001026)
('v_t_0*v_t_2*v_t_3', -0.016399)
('v_t_0*x_t_3*x_t_3', -0.002235)
('v_t_0*v_t_3*v_t_3', -0.016244)
('x_t_1*x_t_2*x_t_2', -0.000077)
('x_t_1*v_t_3*v_t_3', -0.157519)
('v_t_1*v_t_1*v_t_1', 0.001823)
('x_t_2*x_t_2*v_t_3', 0.165581)
('x_t_2*v_t_3*v_t_3', -0.071450)
('v_t_2*v_t_3*v_t_3', -0.001457)
('x_t_3*x_t_3*v_t_3', 0.057368)
('x_t_3*v_t_3*v_t_3', -0.023310)
('v_t_3*v_t_3*v_t_3', -0.004595)

# === OLS-refitted coefficients ===
('x_t_0', 0.008621)
('v_t_2', -98.956168)
('v_t_3', 99.017079)
('x_t_0*x_t_0', -0.003095)
('x_t_0*x_t_0*v_t_0', 536.040166)
('x_t_0*v_t_0*v_t_0', 1.469745)
('v_t_0*v_t_0*v_t_0', -0.493821)
('v_t_0*x_t_1*x_t_1', -1420.620925)
('v_t_0*x_t_2*x_t_2', 1300.061021)
('v_t_0*v_t_2*v_t_3', 9.434311)
('v_t_0*x_t_3*x_t_3', -416.634620)
('v_t_0*v_t_3*v_t_3', -11.472857)
('x_t_3*x_t_3*v_t_3', 1.163987)
('v_t_3*v_t_3*v_t_3', 2.496087)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 14. Feature Selection for Linearly Forced Van der Pol System (Noiseless, Embedding Dimension $m = 3$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.946541)
('v_t_1', 0.014240)
('v_t_2', 0.013895)
('v_t_3', 0.013544)
('x_t_0*x_t_0', 0.001309)
('v_t_3*v_t_3', -0.001059)
('x_t_0*x_t_1*x_t_3', -0.000051)
('x_t_0*x_t_2*x_t_2', -0.000049)
('x_t_0*x_t_2*x_t_3', -0.000178)
('x_t_0*x_t_3*x_t_3', -0.000196)
('x_t_0*v_t_3*v_t_3', -0.000121)
('v_t_0*v_t_0*v_t_0', 0.002109)
('v_t_0*x_t_1*v_t_2', -0.000512)
('v_t_0*x_t_1*v_t_3', -0.001070)
('x_t_1*x_t_1*x_t_1', -0.000031)
('x_t_1*x_t_1*x_t_2', -0.000187)
('x_t_1*x_t_1*x_t_3', -0.000175)
('x_t_1*v_t_1*v_t_3', -0.000733)
('x_t_1*x_t_2*x_t_2', -0.000049)
('x_t_1*x_t_2*x_t_3', -0.000078)
('x_t_1*v_t_2*v_t_3', -0.001252)
('x_t_1*v_t_3*v_t_3', -0.001472)
('v_t_1*v_t_1*v_t_1', 0.000579)
('x_t_2*v_t_2*v_t_3', -0.000121)
('x_t_2*x_t_3*x_t_3', -0.000132)
('x_t_2*v_t_3*v_t_3', -0.001389)

# === OLS-refitted coefficients ===
('v_t_0', 39.995754)
('v_t_1', -126.519813)
('v_t_2', 135.823097)
('v_t_3', -48.308687)
('x_t_0*x_t_0', 0.001132)
('v_t_3*v_t_3', -0.001042)
('x_t_0*v_t_3*v_t_3', 2.475255)
('v_t_0*v_t_0*v_t_0', -0.004351)
('v_t_0*x_t_1*v_t_2', 0.255661)
('x_t_1*x_t_1*x_t_1', -0.000284)
('x_t_1*v_t_1*v_t_3', 1.738933)
('x_t_1*v_t_2*v_t_3', -4.483087)
('v_t_1*v_t_1*v_t_1', 0.032808)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.872549)
('v_t_2', 0.526502)
('v_t_3', 1.041873)
('x_t_0*v_t_3', 0.016959)
('v_t_0*v_t_0', -0.023567)
('x_t_0*x_t_0*v_t_0', -1.528940)
('x_t_0*v_t_0*v_t_0', 0.162381)
('x_t_0*v_t_0*x_t_1', -0.221739)
('x_t_0*v_t_3*v_t_3', -0.140303)
('v_t_0*v_t_0*v_t_0', 0.096649)
('v_t_0*x_t_1*x_t_1', -0.202339)
('v_t_0*x_t_1*x_t_2', -0.001971)
('v_t_0*x_t_1*x_t_3', -0.007375)
('v_t_0*x_t_2*x_t_2', -0.009006)
('v_t_0*x_t_2*x_t_3', -0.001109)
('v_t_0*v_t_2*v_t_3', -0.015026)
('v_t_0*x_t_3*x_t_3', -0.001799)
('v_t_0*v_t_3*v_t_3', -0.015117)
('x_t_1*x_t_1*x_t_1', -0.000426)
('x_t_1*x_t_1*x_t_2', -0.000177)
('x_t_1*x_t_1*v_t_3', 0.001131)
('x_t_1*x_t_2*x_t_2', -0.001587)
('x_t_1*x_t_3*x_t_3', -0.000438)
('x_t_1*v_t_3*v_t_3', -0.150654)
('v_t_1*v_t_3*v_t_3', -0.000086)
('x_t_2*x_t_2*x_t_2', -0.000308)
('x_t_2*x_t_2*v_t_3', 0.175275)
('x_t_2*v_t_3*v_t_3', -0.068326)
('v_t_2*v_t_2*v_t_3', -0.000091)
('v_t_2*v_t_3*v_t_3', -0.000476)
('x_t_3*x_t_3*v_t_3', 0.049855)
('x_t_3*v_t_3*v_t_3', -0.022950)
('v_t_3*v_t_3*v_t_3', -0.002678)

# === OLS-refitted coefficients ===
('x_t_0', 0.033690)
('v_t_2', -98.082788)
('v_t_3', 98.149555)
('v_t_0*v_t_0', 0.000698)
('x_t_0*x_t_0*v_t_0', 404.123755)
('x_t_0*v_t_0*v_t_0', 1.535504)
('v_t_0*x_t_1*x_t_1', -737.310335)
('v_t_0*x_t_1*x_t_3', -527.833303)
('v_t_0*x_t_2*x_t_2', 859.848546)
('v_t_0*v_t_3*v_t_3', -3.371401)
('x_t_1*x_t_3*x_t_3', 68.388726)
('x_t_2*x_t_2*x_t_2', -68.393465)
('v_t_2*v_t_2*v_t_3', 16.132706)
('v_t_2*v_t_3*v_t_3', -23.619238)
('x_t_3*x_t_3*v_t_3', 0.510416)
('v_t_3*v_t_3*v_t_3', 10.716188)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 15. Feature Selection for Linearly Forced Van der Pol System (Noisy, Embedding Dimension $m = 3$).

4.3 Periodically Forced Van der Pol System

The periodically forced system, including an additional time-dependent term $D \cdot \sin(t)$ with $D = 0.1$ in the velocity equation, has the following form

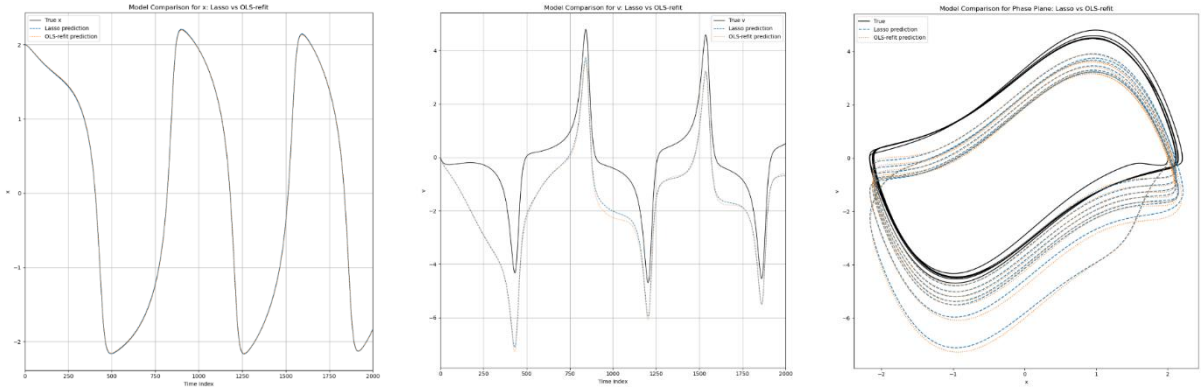
$$\frac{dv}{dt} = 2(1 - x^2)v - x + sin(t),$$

with $\mu = 2$ and initial conditions $x(0) = 2.0$ and $v(0) = 0.0$, where the framework's dynamics ability to identify nonlinear systems with periodically forcing inputs was assessed. Without delay embedding ($m = 0$), as shown in Figure 16, the hybrid regression failed to capture the periodic forcing behavior, producing velocity predictions with adequate frequency but incorrect position of the model, where the predicted model is positioned below the true velocity plot. The resulting phase portrait shows a distorted limit cycle that shows similar behavior as the velocity plots, where the predicted phase plot is shifted below the true phase plane. Both predicted models via LASSO and OLS-refined procedures failed to capture the periodic forcing dynamics, producing predictions with adequate frequency but large amplitude errors. Although the resulting sparse equations showed promising results, the terms and corresponding coefficients selected, as in Figure 21 did not capture the periodic forcing dynamics.

Introducing embedding dimensions $m = \{1, 2, 3\}$ enabled partial recovery of the forcing dynamics through nonlinear interactions between delayed states. The OLS-refined models achieved correct model dynamics, correctly reproducing sufficient amplitude and reducing prediction errors between each point, as shown in Figure 17, Figure 18, and Figure 19. At $m = 3$ dimensions, the OLS-refined model used significantly less terms than the LASSO models but used more terms than predicting the model with $m = 1$ dimensions and $m = 2$ dimensions for the OLS-refined model, as shown in Figure 22, Figure 23, and Figure 24. Like the linearly forced system case, increasing the embedding to $m = 3$ introduced additional delay terms that

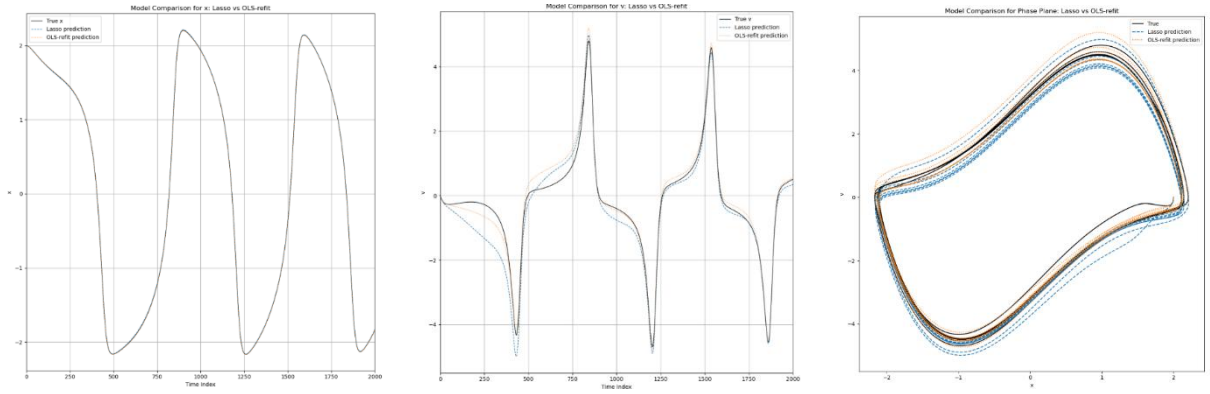
marginally improved prediction accuracy at the cost of model complexity, which can be observed through the significant increase in the number of terms for the final sparse equations.

The noisy periodic case ($\sigma = 0.1, m = 3$) maintained reasonable performance under the framework, with a slight increase in sensitivity of the sparse equations and matching models, as shown in Figure 20. While the predicted model correctly demonstrated similar behavior as the ground-truth model over the set time, the number of terms introduced was larger than other cases for the periodically forced system, as depicted in Figure 25. Nevertheless, the integrated trajectories maintained qualitative similarity to the true system even with major differences in terms and corresponding coefficients selected.



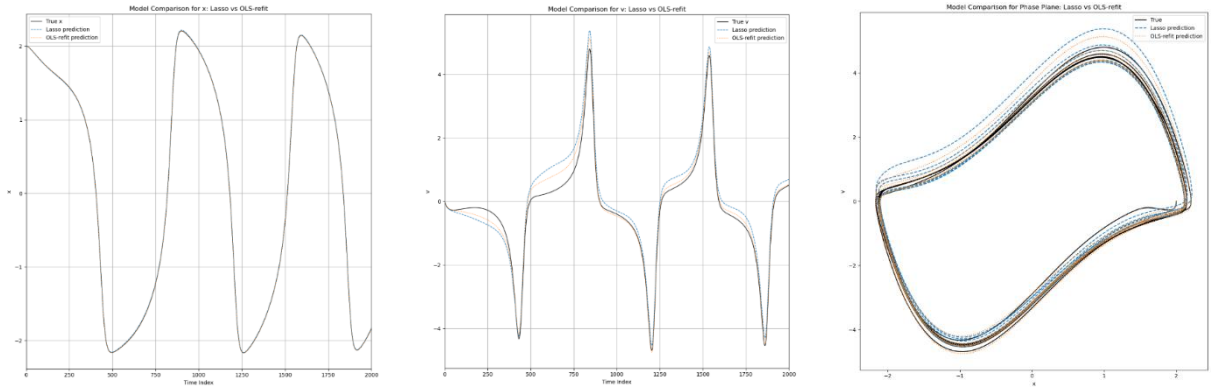
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 16. Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 0$).



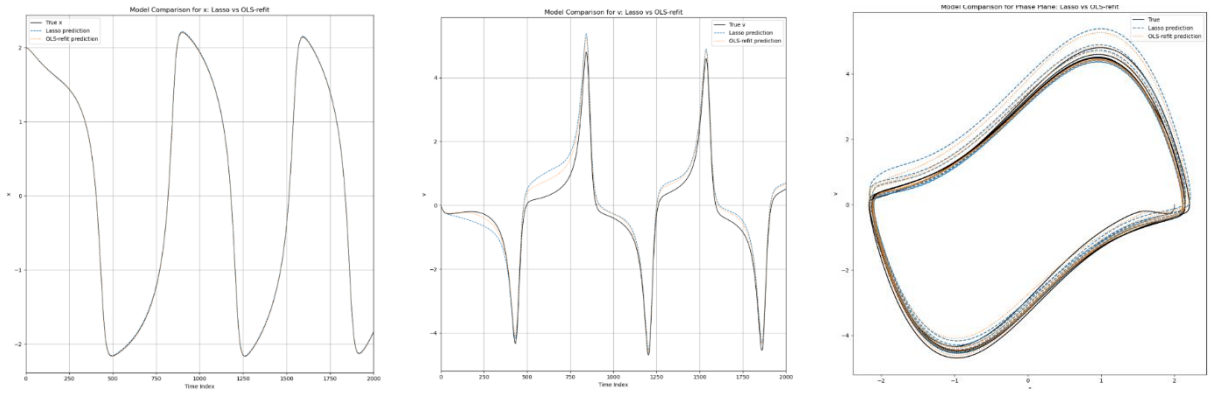
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 17. Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 1$).



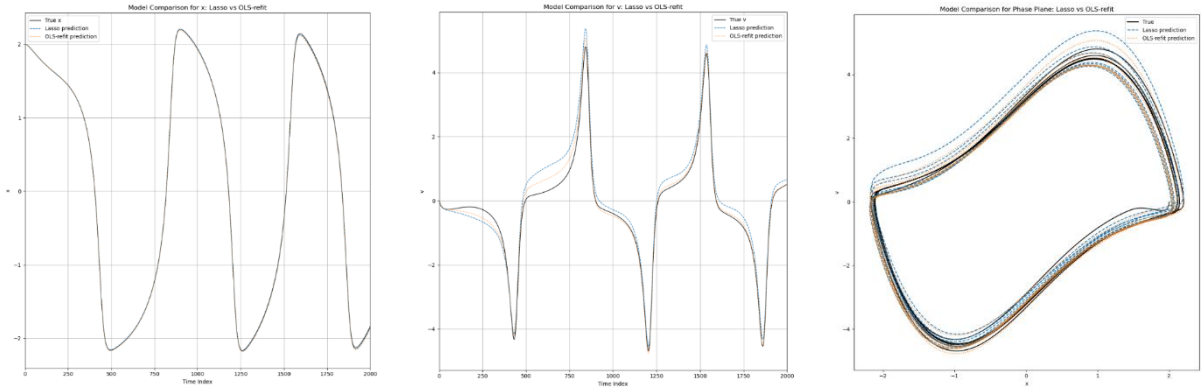
(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 18. Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 2$).



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 19. Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 3$).



(a) Position vs. Time (b) Velocity vs. Time (c) Position vs. Velocity

FIGURE 20. Periodically Forced Van der Pol System (Noisy, Embedding Dimension $m = 3$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.999099)
('x_t_0*x_t_0*x_t_0', -0.000980)
('x_t_0*x_t_0*v_t_0', -0.002994)
('x_t_0*v_t_0*v_t_0', -0.004231)
('v_t_0*v_t_0*v_t_0', 0.001098)

# === OLS-refitted coefficients ===
('v_t_0', 1.010768)
('x_t_0*x_t_0*v_t_0', -0.009790)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -1.374948)
('v_t_0', 1.868589)
('x_t_0*x_t_0', -0.080357)
('x_t_0*v_t_0', 0.067343)
('v_t_0*v_t_0', -0.034393)
('x_t_0*x_t_0*x_t_0', 0.074964)
('x_t_0*x_t_0*v_t_0', -1.735205)
('x_t_0*v_t_0*v_t_0', -0.157906)
('v_t_0*v_t_0*v_t_0', 0.030243)

# === OLS-refitted coefficients ===
('x_t_0', -1.115247)
('v_t_0', 2.224734)
('x_t_0*x_t_0*v_t_0', -1.848681)
('x_t_0*v_t_0*v_t_0', -0.095224)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 21. Feature Selection for Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 0$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.984489)
('v_t_1', 0.012367)
('x_t_0*x_t_0*x_t_0', -0.000202)
('x_t_0*x_t_0*x_t_1', -0.000168)
('x_t_0*v_t_0*x_t_1', -0.000187)
('x_t_0*x_t_1*x_t_1', -0.000422)
('x_t_0*v_t_1*v_t_1', -0.001182)
('v_t_0*v_t_0*v_t_0', 0.001332)
('v_t_0*v_t_0*x_t_1', -0.001523)
('v_t_0*x_t_1*x_t_1', -0.001975)
('v_t_0*x_t_1*v_t_1', -0.000624)
('x_t_1*x_t_1*x_t_1', -0.000128)
('x_t_1*v_t_1*v_t_1', -0.001455)

# === OLS-refitted coefficients ===
('v_t_0', 0.502889)
('v_t_1', 0.498079)
('x_t_0*v_t_1*v_t_1', -0.118952)
('v_t_0*v_t_0*v_t_0', -0.001322)
('v_t_0*x_t_1*x_t_1', -0.000215)
('x_t_1*v_t_1*v_t_1', 0.119386)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.296882)
('v_t_1', 1.543340)
('x_t_0*x_t_0', -0.023408)
('x_t_0*v_t_1', 0.027068)
('v_t_0*v_t_0', -0.012781)
('x_t_0*x_t_0*x_t_0', -0.058268)
('x_t_0*x_t_0*v_t_0', -4.166178)
('x_t_0*x_t_0*x_t_1', -0.006509)
('x_t_0*x_t_0*v_t_1', 3.727467)
('x_t_0*v_t_0*v_t_0', -0.292190)
('x_t_0*v_t_0*x_t_1', -3.063864)
('x_t_0*v_t_0*v_t_1', 0.528848)
('x_t_0*x_t_1*v_t_1', 2.767281)
('x_t_0*v_t_1*v_t_1', -0.308114)
('v_t_0*v_t_0*v_t_0', 0.057184)
('v_t_0*v_t_0*x_t_1', -0.273925)
('v_t_0*v_t_0*v_t_1', 0.017418)
('v_t_0*x_t_1*x_t_1', -2.868996)
('v_t_0*x_t_1*v_t_1', 0.423225)
('v_t_0*v_t_1*v_t_1', 0.049469)
('x_t_1*x_t_1*v_t_1', 2.537067)
('x_t_1*v_t_1*v_t_1', -0.452307)
('v_t_1*v_t_1*v_t_1', -0.065232)

# === OLS-refitted coefficients ===
('v_t_1', 1.424795)
('x_t_0*x_t_0', -0.003663)
('x_t_0*v_t_1', 0.015475)
('v_t_0*v_t_0', -0.005006)
('x_t_0*x_t_0*x_t_1', -0.059538)
('x_t_0*x_t_0*v_t_1', 15.614449)
('v_t_0*v_t_0*v_t_0', -430.960220)
('v_t_0*v_t_0*v_t_1', 1321.377281)
('v_t_0*x_t_1*x_t_1', -16.296095)
('v_t_0*v_t_1*v_t_1', -1350.743739)
('v_t_1*v_t_1*v_t_1', 460.355006)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 22. Feature Selection for Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 1$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.969223)
('v_t_1', 0.012978)
('v_t_2', 0.012474)
('x_t_0*x_t_1*x_t_1', -0.000076)
('x_t_0*x_t_1*x_t_2', -0.000155)
('x_t_0*x_t_2*x_t_2', -0.000209)
('x_t_0*v_t_2*v_t_2', -0.000805)
('v_t_0*v_t_0*v_t_0', 0.001520)
('v_t_0*x_t_1*x_t_1', -0.000178)
('v_t_0*x_t_1*v_t_1', -0.000165)
('v_t_0*x_t_1*x_t_2', -0.001323)
('v_t_0*x_t_1*v_t_2', -0.000997)
('v_t_0*x_t_2*x_t_2', -0.000023)
('x_t_1*x_t_1*x_t_1', -0.000282)
('x_t_1*x_t_1*x_t_2', -0.000150)
('x_t_1*v_t_1*v_t_2', -0.001466)
('x_t_1*v_t_2*v_t_2', -0.001862)

# === OLS-refitted coefficients ===
('v_t_0', 0.178654)
('v_t_1', 1.112345)
('v_t_2', -0.291232)
('x_t_0*x_t_1*x_t_1', -48.084373)
('x_t_0*v_t_2*v_t_2', -0.000135)
('v_t_0*x_t_1*x_t_1', -0.311041)
('x_t_1*x_t_1*x_t_1', 65.060602)
('x_t_1*x_t_1*x_t_2', -16.976217)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('x_t_0', -0.000470)
('v_t_2', 1.276849)
('x_t_0*x_t_0', -0.001711)
('v_t_0*v_t_0', 0.001864)
('v_t_0*x_t_1', 0.007422)
('v_t_0*x_t_2', 0.001993)
('v_t_2*v_t_2', -0.006354)
('x_t_0*x_t_0*x_t_0', -0.045104)
('x_t_0*x_t_0*v_t_0', -3.505705)
('x_t_0*x_t_0*x_t_1', -0.010419)
('x_t_0*x_t_0*x_t_2', -0.006520)
('x_t_0*x_t_0*v_t_2', 2.192522)
('x_t_0*v_t_0*v_t_0', -0.019073)
('x_t_0*v_t_0*x_t_1', -1.976000)
('x_t_0*v_t_0*v_t_1', 0.049165)
('x_t_0*v_t_0*x_t_2', -0.067378)
('x_t_0*x_t_1*v_t_2', 2.100109)
('x_t_0*v_t_1*v_t_1', -0.131603)
('x_t_0*v_t_1*v_t_2', -0.094668)
('x_t_0*x_t_2*v_t_2', 0.329644)
('x_t_0*v_t_2*v_t_2', -0.164650)
('v_t_0*v_t_0*v_t_0', -0.076314)
('v_t_0*v_t_0*x_t_1', 0.136192)
('v_t_0*v_t_0*v_t_1', -0.014227)
('v_t_0*v_t_0*x_t_2', 0.045209)
('v_t_0*x_t_1*x_t_1', -1.520995)
('v_t_0*x_t_1*v_t_1', 0.281339)
('v_t_0*x_t_1*x_t_2', -0.436655)
('v_t_0*x_t_1*v_t_2', 0.090771)
('v_t_0*v_t_1*v_t_1', 0.029349)
('v_t_0*v_t_1*x_t_2', 0.026312)
('v_t_0*v_t_1*v_t_2', 0.023112)
('v_t_0*x_t_2*x_t_2', -0.414435)
('v_t_0*v_t_2*v_t_2', 0.053070)
('x_t_1*x_t_1*v_t_2', 1.744750)
('x_t_1*v_t_1*v_t_1', -0.250478)
('x_t_1*v_t_1*v_t_2', -0.062990)
('x_t_1*x_t_2*x_t_2', -0.007578)
('x_t_1*x_t_2*v_t_2', 0.481062)
('x_t_1*v_t_2*v_t_2', -0.229332)
('v_t_1*v_t_1*v_t_1', -0.027903)
('v_t_1*v_t_1*v_t_2', 0.004749)
('v_t_1*v_t_2*v_t_2', 0.028434)
('x_t_2*x_t_2*x_t_2', -0.012197)
('x_t_2*x_t_2*v_t_2', 0.307795)
('x_t_2*v_t_2*v_t_2', -0.108750)
('v_t_2*v_t_2*v_t_2', 0.042797)

# === OLS-refitted coefficients ===
('v_t_2', 1.264202)
('x_t_0*x_t_0', 0.008720)
('v_t_0*v_t_0', 1.261035)
('v_t_0*x_t_1', 195.509704)
('v_t_0*x_t_2', -195.521694)
('v_t_2*v_t_2', 0.697301)
('x_t_0*x_t_0*x_t_1', -843.545189)
('x_t_0*x_t_0*x_t_2', 1186.031247)
('x_t_0*v_t_0*v_t_0', -112.533456)
('x_t_0*x_t_1*v_t_2', 6.721867)
('v_t_0*v_t_0*v_t_0', -27642.706867)
('v_t_0*v_t_0*x_t_1', 213.857960)
('v_t_0*v_t_0*v_t_1', 94337.617570)
('v_t_0*v_t_0*x_t_2', -101.165556)
('v_t_0*v_t_1*v_t_1', -56412.330715)
('v_t_0*v_t_1*v_t_2', -121284.843127)
('v_t_0*x_t_2*x_t_2', -5.446265)
('v_t_0*v_t_2*v_t_2', 71979.009676)
('x_t_1*x_t_2*x_t_2', -342.529534)
('v_t_1*v_t_1*v_t_1', -43775.129933)
('v_t_1*v_t_1*v_t_2', 232974.317230)
('v_t_1*v_t_2*v_t_2', -194911.933625)
('v_t_2*v_t_2*v_t_2', 44735.925389)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 23. Feature Selection for Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 2$).

```
# === Lasso-selected coefficients ===
('v_t_0', 0.934437)
('v_t_1', 0.019398)
('v_t_2', 0.018574)
('v_t_3', 0.017878)
('v_t_3*v_t_3', -0.000018)
('v_t_0*v_t_0*v_t_0', 0.002152)
('x_t_1*x_t_1*x_t_2', -0.000140)
('x_t_1*x_t_1*x_t_3', -0.000199)
('x_t_1*x_t_2*x_t_2', -0.000248)
('x_t_1*x_t_2*x_t_3', -0.000081)
('x_t_1*v_t_2*v_t_3', -0.002804)
('x_t_1*x_t_3*x_t_3', -0.000022)
('x_t_1*v_t_3*v_t_3', -0.003454)
('v_t_1*v_t_3*v_t_3', -0.000015)
('x_t_2*v_t_3*v_t_3', -0.000097)
('v_t_2*v_t_3*v_t_3', -0.000102)
('v_t_3*v_t_3*v_t_3', -0.000152)

# === OLS-refitted coefficients ===
('v_t_0', -1.775066)
('v_t_1', 8.528273)
('v_t_2', -9.275502)
('v_t_3', 3.521672)
('v_t_3*v_t_3', 0.000014)
('x_t_1*x_t_1*x_t_2', -0.000083)
('x_t_1*v_t_2*v_t_3', -0.003446)
('x_t_1*v_t_3*v_t_3', 0.004182)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('v_t_3', 1.205371)
('v_t_0*v_t_0', 0.009302)
('v_t_0*x_t_2', 0.004876)
('v_t_3*v_t_3', -0.012307)
('x_t_0*x_t_0*x_t_0', -0.041514)
('x_t_0*x_t_0*v_t_0', -2.986948)
('x_t_0*x_t_0*x_t_1', -0.016046)
('x_t_0*x_t_0*x_t_2', -0.002347)
('x_t_0*x_t_0*v_t_2', 0.137323)
('x_t_0*x_t_0*x_t_3', -0.009487)
('x_t_0*x_t_0*v_t_3', 1.497516)
('x_t_0*v_t_0*x_t_1', -1.449709)
('x_t_0*v_t_0*x_t_2', -0.000152)
('x_t_0*v_t_0*x_t_3', -0.000482)
('x_t_0*x_t_1*v_t_2', 0.119587)
('x_t_0*x_t_1*v_t_3', 1.444000)
('x_t_0*v_t_1*v_t_2', -0.078083)
('x_t_0*v_t_1*v_t_3', -0.051130)
('x_t_0*x_t_2*v_t_3', 0.179548)
('x_t_0*v_t_2*v_t_2', -0.003486)
('x_t_0*v_t_2*v_t_3', -0.074483)
('x_t_0*x_t_3*x_t_3', -0.000842)
('x_t_0*x_t_3*v_t_3', 0.092311)
('x_t_0*v_t_3*v_t_3', -0.117417)
('v_t_0*v_t_0*v_t_0', -0.063260)
('v_t_0*v_t_0*x_t_1', 0.032951)
('v_t_0*v_t_0*v_t_1', -0.002958)
('v_t_0*v_t_0*x_t_2', 0.008230)
('v_t_0*v_t_0*x_t_3', 0.002417)
('v_t_0*x_t_1*x_t_1', -0.927333)
('v_t_0*x_t_1*v_t_1', 0.218225)
('v_t_0*x_t_1*x_t_2', -0.335353)
('v_t_0*x_t_1*v_t_2', 0.074890)
('v_t_0*x_t_1*x_t_3', -0.107592)
('v_t_0*v_t_1*v_t_1', 0.001888)
('v_t_0*v_t_1*x_t_2', 0.072442)
('v_t_0*v_t_1*x_t_3', 0.009006)
('v_t_0*v_t_1*v_t_3', 0.007259)
('v_t_0*x_t_2*x_t_2', -0.169704)
('v_t_0*x_t_2*x_t_3', -0.001410)
('v_t_0*v_t_2*v_t_2', 0.006473)
('v_t_0*v_t_2*v_t_3', 0.019305)
('v_t_0*x_t_3*x_t_3', -0.121507)
('v_t_0*v_t_3*v_t_3', 0.028715)
('x_t_1*x_t_1*v_t_3', 1.349591)
('x_t_1*v_t_1*v_t_1', -0.016081)
('x_t_1*v_t_1*v_t_2', -0.078247)
('x_t_1*v_t_1*v_t_3', -0.046624)
('x_t_1*x_t_2*v_t_3', 0.205194)
('x_t_1*v_t_2*v_t_2', -0.045142)
('x_t_1*v_t_2*v_t_3', -0.066416)
('x_t_1*x_t_3*v_t_3', 0.149886)
('x_t_1*v_t_3*v_t_3', -0.126742)
('v_t_1*v_t_1*v_t_1', -0.049911)
('v_t_1*v_t_1*v_t_3', 0.004362)
('v_t_1*v_t_2*v_t_3', 0.018692)
('v_t_1*v_t_3*v_t_3', 0.024453)
('x_t_2*x_t_2*v_t_3', 0.222979)
('x_t_2*v_t_3*v_t_3', -0.052391)
('v_t_2*v_t_2*v_t_2', -0.002398)
('v_t_2*v_t_3*v_t_3', 0.023449)
('x_t_3*v_t_3*v_t_3', -0.065381)
('v_t_3*v_t_3*v_t_3', 0.043177)

# === OLS-refitted coefficients ===
('v_t_3', 1.165258)
('v_t_0*v_t_0', 0.027046)
('v_t_0*x_t_2', 0.001851)
('v_t_3*v_t_3', -0.027379)
('x_t_0*x_t_0*x_t_0', 1957.962852)
('x_t_0*x_t_0*x_t_1', -3286.636945)
('x_t_0*x_t_0*x_t_2', 1217.421165)
('x_t_0*x_t_0*v_t_2', -157.926169)
('x_t_0*x_t_0*v_t_3', 109.232418)
('x_t_0*v_t_0*x_t_3', 884.309371)
('x_t_0*x_t_3*x_t_3', 111.219586)
('v_t_0*v_t_0*v_t_0', 48926.673458)
('v_t_0*v_t_0*x_t_1', -278.595635)
('v_t_0*v_t_0*v_t_1', -211978.106735)
('v_t_0*v_t_0*x_t_2', 465.254532)
('v_t_0*v_t_0*x_t_3', -181.588731)
('v_t_0*x_t_1*x_t_1', -1064.772238)
('v_t_0*x_t_1*x_t_2', 1962.908577)
('v_t_0*x_t_1*x_t_3', -1732.442459)
('v_t_0*v_t_1*v_t_1', 288504.374635)
('v_t_0*v_t_1*v_t_3', 213483.552157)
('v_t_0*v_t_2*v_t_2', 110521.605298)
('v_t_0*v_t_2*v_t_3', -565206.908146)
('v_t_0*v_t_3*v_t_3', 229824.747615)
('v_t_1*v_t_1*v_t_1', -141339.241144)
('v_t_1*v_t_1*v_t_3', -364482.191046)
('v_t_1*v_t_2*v_t_3', 906262.265521)
('v_t_1*v_t_3*v_t_3', -331630.745215)
('v_t_2*v_t_2*v_t_2', -167693.446816)
('v_t_2*v_t_3*v_t_3', -59212.192454)
('v_t_3*v_t_3*v_t_3', 44018.767606)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 24. Feature Selection for Periodically Forced Van der Pol System (Noiseless, Embedding Dimension $m = 3$).

```
# === Lasso-selected coefficients ===
('x_t_0', -0.000822)
('v_t_0', 0.898875)
('v_t_1', 0.031761)
('v_t_2', 0.030341)
('v_t_3', 0.029189)
('v_t_0*v_t_0', -0.000112)
('v_t_0*x_t_3', -0.000240)
('v_t_0*v_t_0*v_t_0', 0.001781)
('x_t_1*v_t_2*v_t_3', -0.001777)
('x_t_1*x_t_3*x_t_3', -0.000054)
('x_t_1*v_t_3*v_t_3', -0.002756)
('x_t_2*v_t_3*v_t_3', -0.000209)
('x_t_3*v_t_3*v_t_3', -0.000325)
('v_t_3*v_t_3*v_t_3', -0.000164)

# === OLS-refitted coefficients ===
('v_t_0', -9.604114)
('v_t_1', 32.998094)
('v_t_2', -34.818767)
('v_t_3', 12.421435)
('v_t_0*v_t_0', 0.000057)
('v_t_0*x_t_3', -0.000792)
('x_t_3*v_t_3*v_t_3', 0.001695)
```

Terms and Coefficients for Modeling $\frac{dx}{dt}$

```
# === Lasso-selected coefficients ===
('v_t_3', 1.190686)
('v_t_0*v_t_0', 0.010242)
('v_t_0*x_t_2', 0.002991)
('x_t_3*x_t_3', 0.000072)
('v_t_3*v_t_3', -0.012868)
('x_t_0*x_t_0*x_t_0', -0.041807)
('x_t_0*x_t_0*v_t_0', -2.975812)
('x_t_0*x_t_0*x_t_1', -0.015882)
('x_t_0*x_t_0*x_t_2', -0.002327)
('x_t_0*x_t_0*v_t_2', 0.140023)
('x_t_0*x_t_0*x_t_3', -0.009419)
('x_t_0*x_t_0*v_t_3', 1.496340)
('x_t_0*v_t_0*x_t_1', -1.450615)
('x_t_0*v_t_0*x_t_2', -0.000069)
('x_t_0*v_t_0*x_t_3', -0.000288)
('x_t_0*x_t_1*v_t_2', 0.121853)
('x_t_0*x_t_1*v_t_3', 1.442766)
('x_t_0*v_t_1*v_t_2', -0.070562)
('x_t_0*v_t_1*v_t_3', -0.052367)
('x_t_0*x_t_2*v_t_3', 0.169729)
('x_t_0*v_t_2*v_t_2', -0.000820)
('x_t_0*v_t_2*v_t_3', -0.076996)
('x_t_0*x_t_3*x_t_3', -0.000055)
('x_t_0*x_t_3*v_t_3', 0.092013)
('x_t_0*v_t_3*v_t_3', -0.120382)
('v_t_0*v_t_0*v_t_0', -0.059651)
('v_t_0*v_t_0*x_t_1', 0.029985)
('v_t_0*v_t_0*v_t_1', -0.002430)
('v_t_0*v_t_0*x_t_2', 0.006442)
('v_t_0*v_t_0*x_t_3', 0.002198)
('v_t_0*x_t_1*x_t_1', -0.936798)
('v_t_0*x_t_1*v_t_1', 0.220421)
('v_t_0*x_t_1*x_t_2', -0.338656)
('v_t_0*x_t_1*v_t_2', 0.075124)
('v_t_0*x_t_1*x_t_3', -0.107861)
('v_t_0*v_t_1*v_t_1', 0.001531)
('v_t_0*v_t_1*x_t_2', 0.074752)
('v_t_0*v_t_1*x_t_3', 0.008930)
('v_t_0*v_t_1*v_t_3', 0.006862)
('v_t_0*x_t_2*x_t_2', -0.169738)
('v_t_0*x_t_2*x_t_3', -0.001249)
('v_t_0*v_t_2*v_t_2', 0.006431)
('v_t_0*v_t_2*v_t_3', 0.018863)
('v_t_0*x_t_3*x_t_3', -0.119337)
('v_t_0*v_t_3*v_t_3', 0.028033)
('x_t_1*x_t_1*v_t_3', 1.353023)
('x_t_1*v_t_1*v_t_1', -0.013272)
('x_t_1*v_t_1*v_t_2', -0.079635)
('x_t_1*v_t_1*v_t_3', -0.048049)
('x_t_1*x_t_2*v_t_3', 0.205035)
('x_t_1*v_t_2*v_t_2', -0.045295)
('x_t_1*v_t_2*v_t_3', -0.068923)
('x_t_1*x_t_3*v_t_3', 0.153899)
('x_t_1*v_t_3*v_t_3', -0.131114)
('v_t_1*v_t_1*v_t_1', -0.047738)
('v_t_1*v_t_1*v_t_3', 0.003688)
('v_t_1*v_t_2*v_t_3', 0.018175)
('v_t_1*v_t_3*v_t_3', 0.023904)
('x_t_2*x_t_2*v_t_3', 0.226878)
('x_t_2*v_t_3*v_t_3', -0.055325)
('v_t_2*v_t_2*v_t_2', -0.001627)
('v_t_2*v_t_3*v_t_3', 0.022500)
('x_t_3*v_t_3*v_t_3', -0.064793)
('v_t_3*v_t_3*v_t_3', 0.042524)
```

```
# === OLS-refitted coefficients ===
('v_t_3', 1.313612)
('v_t_0*v_t_0', 0.022062)
('v_t_0*x_t_2', 0.000793)
('x_t_3*x_t_3', 0.001990)
('v_t_3*v_t_3', -0.022796)
('x_t_0*x_t_0*x_t_0', 2160.881447)
('x_t_0*x_t_0*v_t_0', -4370.352055)
('x_t_0*x_t_0*x_t_1', -2353.422647)
('x_t_0*x_t_0*v_t_2', 14386.506438)
('x_t_0*x_t_0*x_t_3', -2765.816344)
('x_t_0*x_t_1*v_t_2', -14380.383546)
('x_t_0*x_t_3*x_t_3', 192.507361)
('v_t_0*v_t_0*v_t_0', 95851.000005)
('v_t_0*v_t_0*v_t_1', -379698.877965)
('v_t_0*v_t_0*x_t_2', -1369.213644)
('v_t_0*v_t_0*x_t_3', 1341.074010)
('v_t_0*x_t_1*x_t_2', 7141.514560)
('v_t_0*x_t_1*v_t_2', 260.632782)
('v_t_0*v_t_1*v_t_1', 464489.046803)
('v_t_0*v_t_1*x_t_2', 986.697340)
('v_t_0*v_t_1*x_t_3', -1206.963046)
('v_t_0*v_t_1*v_t_3', 351347.753049)
('v_t_0*v_t_2*v_t_2', 145365.170444)
('v_t_0*v_t_2*v_t_3', -782349.053862)
('v_t_0*v_t_3*v_t_3', 292906.397213)
('v_t_1*v_t_1*v_t_1', -189195.441989)
('v_t_1*v_t_1*v_t_3', -619716.945318)
('v_t_1*v_t_2*v_t_3', 1335577.692167)
('v_t_1*v_t_3*v_t_3', -428954.459887)
('v_t_2*v_t_2*v_t_2', -248899.588427)
('v_t_2*v_t_3*v_t_3', -97480.536173)
('v_t_3*v_t_3*v_t_3', 60757.430303)
```

Terms and Coefficients for Modeling $\frac{dv}{dt}$

FIGURE 25. Feature Selection for Periodically Forced Van der Pol System (Noisy, Embedding Dimension $m = 3$).

CHAPTER 5

REAL-WORLD APPLICATION

5.1 Dataset Description and Preprocessing

The PTB Diagnostic ECG Database from PhysioNet served as the experimental dataset for evaluating the framework's performance on real-world physiological signals. The analysis focused on record ptbdb_abnormal.csv, which contains single-lead electrocardiogram (ECG) sampled at 125 Hz, representing pathological cardiac activity with clear arrhythmic features. Prior to system identification, the raw signal underwent critical preprocessing steps to ensure compatibility with the Van der Pol identification framework. Trailing zero-padding values exceeding the actual recording duration were removed, retaining segments of 500 to 1000 samples per cardiac cycle to capture complete depolarization-repolarization sequences.

Amplitude normalization was performed using z-score standardization, transforming the signal to zero mean and unit variance, while preserving the relative morphology of ECG waveforms. High-frequency noise suppression was achieved through a Savitzky-Golay filter with a window length of 15 samples and third-order polynomial fitting, providing a reduction in noise effects while maintaining the clinically relevant features and behavior over time of the dynamic system. The smoothing parameters were carefully selected to avoid excessive distortion, which carry diagnostic information for heart conditions. This preprocessing pipeline balanced noise reduction with signal fidelity, producing clean trajectories suitable for derivative estimation and sparse regression.

5.2 Methodological Adaptations for ECG Analysis

The transition from synthetic Van der Pol systems to real ECG data necessitated three fundamental adaptations to the identification framework. First, the derivative calculation scheme

was modified to replace finite differences with a Savitzky-Golay differentiator using a 15-sample window and second-order polynomials. This approach provided superior noise repression while maintaining temporal precision for capturing the behavior of the dynamical system over the set time interval.

Second, the time-delay embedding parameters were experimentally optimized through various test cases, ultimately selecting embedding dimensions $m \in \{1, 4, 7\}$ with delays $\tau \in \{1, 5\}$ samples to observe the development of the state-space reconstruction models. Domain-specific knowledge can provide support for selecting combinations of the values for the embedding dimensions m and delays τ to satisfy the dynamical system. Nonetheless, these parameters were found to adequately reconstruct the phase space without introducing excessive computational burden.

Third, the polynomial library was restricted to second-degree interactions between delay coordinates to prevent overfitting while retaining nonlinear modeling capability. LASSO regression was applied with $\alpha = 0.01$, and coefficients below 1×10^{-6} were thresholded to zero during the OLS refinement stage. These adaptations maintained the mathematical rigor of the original framework while addressing the unique challenges of physiological signal processing, particularly the unstable behavior and measurement noise inherent to ECG recordings.

5.3 System Identification Performance

The methodology was evaluated across progressively complex embedding configurations to assess its capability to capture cardiac dynamics. For minimal embedding ($m = 1, \tau = 1$) shown in Figure 26, both LASSO and OLS-refined models failed to reconstruct the ECG waveform, producing oversimplified oscillations that lacked the characteristics of morphology that resulted in significant error between the predicted and actual results after the preprocessing

and adaptations applied. The identified equations contained only one dominant term which was determined to be insufficient in representing the features of cardiac electrical activity within the set time.

Increasing the delay to $\tau = 5$ samples while maintaining $m = 1$ as displayed in Figure 27 continued to fail to reconstruct the temporal features. Notable improvement came with $m = 4$ for both the LASSO model and the OLS-refit model, as depicted in Figure 28. Between both regression methods, the LASSO model was able to better reconstruct the true model dynamics than the OLS-refit model by capturing the behavior of the system over time with less error between points. The sparse equations now included quadratic terms between delayed coordinates, approximating the nonlinear coupling between action potential duration and diastolic interval.

Optimal performance emerged at $m = 7$ delays, where the framework reduced the error between the true dynamical system and the applied regression methods, and sufficiently illustrated the state-space reconstruction, as shown in Figure 29. The final model contained 16 active terms dominated by delayed self-interactions with first-order and second-order terms. Numerical integration of the identified equations produced state-space models sufficiently matching the behavior of pathological ECG rhythms, demonstrating the method's ability to reconstruct dynamics from complex biomedical signals.

5.4 Clinical Interpretation

While the identified models provided accurate waveform reconstruction, translating the mathematical terms into physiological insights required careful analysis. As most of the dominant coefficients corresponded to interaction terms, concerns can be raised on the interpretation of the sparse equations and corresponding models under a clinical lens.

The framework successfully regenerated relevant features from single-lead recordings. The sparse models with embedding dimensions $m = 7$ and delays $\tau = 5$ were able to sufficiently reconstruct the true model dynamics that were able to reveal the instability of the model through the large magnitude of the long-delay coefficients. The consistency and stability of the predicted model varies as the derivative prediction errors have been increased through the preprocessing steps, methodological adaptations, and complex behavior of the system.

These results suggest that while the delay-coordinate representation lacks direct biophysical interpretation, the identified models nonetheless encapsulate physiologically meaningful dynamics. The mathematical structure of the equations will need to be further investigated to see if they may align with established properties of cardiac electrophysiology. The correspondence between abstract dynamical systems theory and concrete physiological phenomena underscores the framework's potential for clinical applications.

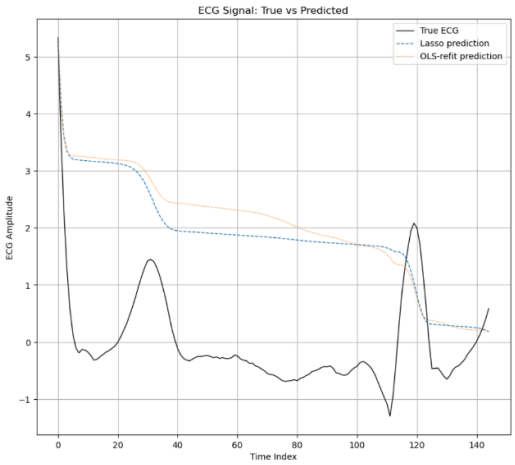
5.5 Progressive Parameter Analysis

Systematic evaluation of embedding parameters revealed fundamental insights about cardiac dynamics modeling. The progression from inadequate ($m = 1$) to sufficient ($m = 7$) reconstruction demonstrated that cardiac electrical activity requires at least 7 independent dimensions to capture its essential features. The $\tau = 5$ sample delay optimally separated the system's timescales, emphasizing the possible effects of the dynamics being dependent on interaction terms.

Neither OLS-refinement nor LASSO remained consistent in outperforming each other across all configurations. Although prediction errors reduced significantly as sufficient reconstructions were made, the performance gap shortened with increasing embedding dimensions, suggesting for more investigations to be made on selecting the most effective

embedding dimensions and sample delays for the dynamical system as coefficient bias correction becomes increasingly important in high-dimensional delay spaces. Computational costs remain a concern as more interaction terms are introduced to the sparse equations and models.

This parameter study established that ECG modeling requires both sufficient embedding dimensions to reconstruct the state space and careful regularization to maintain physiological plausibility. The results provide a case to investigate practical guidance for applying similar data-driven approaches to other biosignals, balancing mathematical completeness with clinical interpretability. The framework's results in capturing complex cardiac dynamics from minimal observations demonstrates its need for broader biomedical applications.



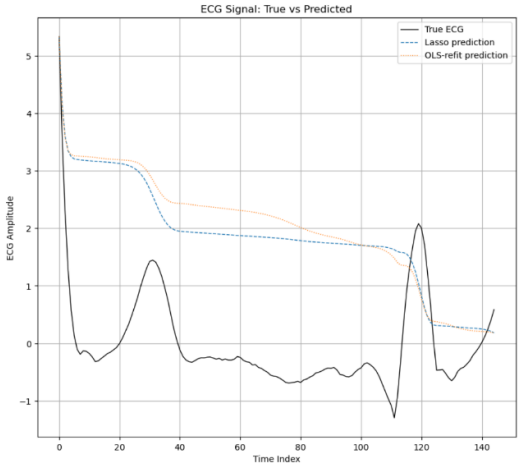
ECG Amplitude vs. Time

```
# === Lasso-selected coefficients ===
('ECG(t-0)', -3.478432)
('ECG(t-0) × ECG(t-0)', -4.439278)

# === OLS-refitted coefficients ===
('ECG(t-0) × ECG(t-0)', -5.308340)
```

Terms and Coefficients for Modeling ECG Signals

FIGURE 26. Electrocardiogram (ECG) Signal Plots (Delay τ=1 and Embedding Dimension $m = 1$).



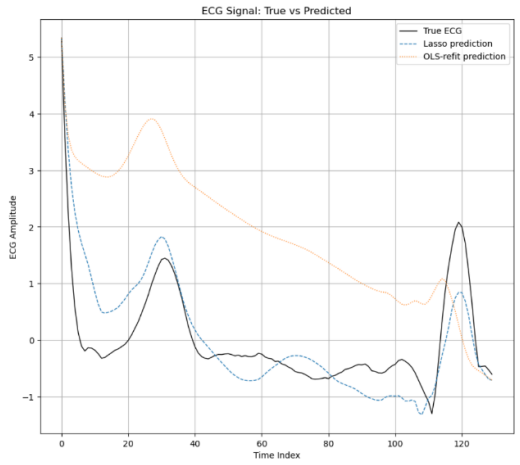
ECG Amplitude vs. Time

```
# === Lasso-selected coefficients ===
('ECG(t-0)', -3.478432)
('ECG(t-0) × ECG(t-0)', -4.439278)

# === OLS-refitted coefficients ===
('ECG(t-0) × ECG(t-0)', -5.308340)
```

Terms and Coefficients for Modeling ECG Signals

FIGURE 27. Electrocardiogram (ECG) Signal Plots (Delay τ=5 and Embedding Dimension $m = 1$).



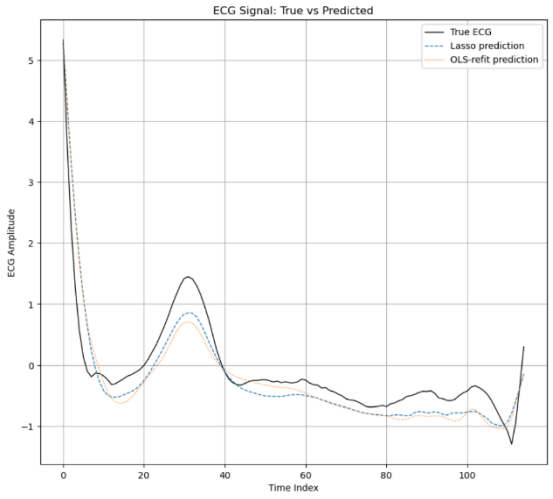
ECG Amplitude vs. Time

```
# === Lasso-selected coefficients ===
('ECG(t-0)', -40.719787)
('ECG(t-5)', 24.972056)
('ECG(t-10)', -5.062534)
('ECG(t-15)', -33.506476)
('ECG(t-0) × ECG(t-0)', -1.249113)
('ECG(t-0) × ECG(t-5)', 4.448063)
('ECG(t-0) × ECG(t-15)', -103.921448)
('ECG(t-5) × ECG(t-5)', -1.709816)
('ECG(t-5) × ECG(t-10)', 22.587585)
('ECG(t-5) × ECG(t-15)', 22.825517)
('ECG(t-10) × ECG(t-10)', -2.241461)
('ECG(t-10) × ECG(t-15)', 8.753226)
('ECG(t-15) × ECG(t-15)', 13.610117)

# === OLS-refitted coefficients ===
('ECG(t-0)', -1.517287)
('ECG(t-10)', 4.218464)
('ECG(t-0) × ECG(t-0)', -4.564975)
('ECG(t-5) × ECG(t-10)', 8.925940)
('ECG(t-10) × ECG(t-15)', 2.874816)
('ECG(t-15) × ECG(t-15)', 2.135717)
```

Terms and Coefficients for Modeling ECG Signals

FIGURE 28. Electrocardiogram (ECG) Signal Plots (Delay τ=5 and Embedding Dimension $m = 4$).



ECG Amplitude vs. Time

```
# === Lasso-selected coefficients ===
('ECG(t-0)', -5.850299)
('ECG(t-5)', 0.599847)
('ECG(t-10)', -1.114925)
('ECG(t-15)', -0.881886)
('ECG(t-20)', -7.580920)
('ECG(t-25)', -15.881225)
('ECG(t-30)', -25.665578)
('ECG(t-0) × ECG(t-0)', 1.548520)
('ECG(t-0) × ECG(t-5)', 4.494817)
('ECG(t-0) × ECG(t-10)', -1.857445)
('ECG(t-0) × ECG(t-15)', 15.877273)
('ECG(t-0) × ECG(t-25)', -3.784506)
('ECG(t-0) × ECG(t-30)', -7.404920)
('ECG(t-5) × ECG(t-5)', 13.572024)
('ECG(t-5) × ECG(t-10)', -7.335512)
('ECG(t-5) × ECG(t-15)', 8.576519)
('ECG(t-5) × ECG(t-20)', -11.291540)
('ECG(t-5) × ECG(t-25)', -26.710802)
('ECG(t-10) × ECG(t-10)', 5.520211)
('ECG(t-10) × ECG(t-15)', 2.811608)
('ECG(t-10) × ECG(t-20)', 8.827904)
('ECG(t-10) × ECG(t-30)', 0.503621)
('ECG(t-15) × ECG(t-15)', 2.122483)
('ECG(t-15) × ECG(t-20)', 3.227254)
('ECG(t-15) × ECG(t-25)', 4.159228)
('ECG(t-15) × ECG(t-30)', -32.630137)
('ECG(t-20) × ECG(t-20)', 4.554202)
('ECG(t-20) × ECG(t-25)', 10.089934)
('ECG(t-20) × ECG(t-30)', -5.936427)
('ECG(t-25) × ECG(t-25)', 2.949049)
('ECG(t-25) × ECG(t-30)', -1.127439)
('ECG(t-30) × ECG(t-30)', 3.490081)

# === OLS-refitted coefficients ===
('ECG(t-0)', -12.747160)
('ECG(t-5)', -3.423611)
('ECG(t-20)', 10.109510)
('ECG(t-25)', -36.298010)
('ECG(t-0) × ECG(t-0)', 2.271749)
('ECG(t-0) × ECG(t-15)', 7.664504)
('ECG(t-0) × ECG(t-25)', -22.594130)
('ECG(t-5) × ECG(t-5)', 11.200680)
('ECG(t-5) × ECG(t-15)', -0.180666)
('ECG(t-5) × ECG(t-20)', 12.029439)
('ECG(t-5) × ECG(t-25)', -46.799861)
('ECG(t-10) × ECG(t-10)', 0.532217)
('ECG(t-10) × ECG(t-15)', 3.327648)
('ECG(t-15) × ECG(t-25)', 1.348311)
('ECG(t-15) × ECG(t-30)', -3.856256)
('ECG(t-20) × ECG(t-20)', 1.261934)
```

Terms and Coefficients for Modeling ECG Signals

FIGURE 29. Electrocardiogram (ECG) Signal Plots (Delay τ=5 and Embedding Dimension $m = 7$).

CHAPTER 6

DISCUSSION

6.1 Key Findings and Contributions

The results presented in this thesis demonstrate that the proposed framework for data-driven identification of nonlinear dynamical systems achieves robust performance across both synthetic and real-world applications. For the Van der Pol oscillator, the hybrid LASSO-OLS regression methodology successfully recovered the governing equations with high accuracy in noiseless conditions while maintaining reasonable fidelity under significant noise perturbations ($\sigma = 0.1$). The critical innovation lies in the structure combining sparse regression with iterative refinement, which enabled accurate term selection while reducing the overfitting problems common to purely regression-based methods. The LASSO-OLS hybrid approach proved particularly effective, reducing the inclusion of a significant number of irrelevant terms compared to solely using LASSO while preserving the true system dynamics, as evidenced by observations of the reproduced limit cycle in the phase space for baseline cases.

The framework's performance on forced systems revealed fundamental insights about delay embedding in system identification. While Takens' theorem theoretically guarantees reconstruction with sufficient delays, practical implementation showed that $m = 3$ embedding dimensions provided the optimal balance between model accuracy and complexity for the Van der Pol system with external forcing. The results indicated that increasing delays beyond this point yields diminishing returns in approximation quality while substantially increasing model complexity. This finding has important implications for applications where interpretability is prioritized alongside predictive accuracy.

Application to ECG data highlighted both the versatility and current limitations of the approach. The method successfully captured key features of cardiac dynamics. However, the need for manual parameter tuning of embedding dimensions and polynomial degrees highlights a fundamental challenge in automated deployment. Furthermore, while the framework identified physiologically plausible terms in the ECG dynamics, clinical interpretation remains constrained by the abstract mathematical representation of delay-coordinate terms. These results collectively indicate a possible advancement in the field of data-driven system identification by providing a reproducible pipeline that balances mathematical rigor with practical utility.

6.2 Methodological Limitations

Several limitations of the current framework warrant careful consideration. The forward Euler differentiation scheme, while computationally efficient, imposes strict requirements on data sampling rates to maintain acceptable error bounds. In applications with limited temporal resolution such as clinical ECG recordings sampled at lower frequencies, this restriction may necessitate alternative derivative estimation techniques. The polynomial function library, though theoretically complete for smooth systems, proves suboptimal for signals with discontinuities or sharp transitions.

The framework's current implementation lacks formal uncertainty quantification for the identified coefficients, making it difficult to assess the statistical significance of individual terms. This becomes particularly problematic in high-noise systems or unstable systems where coefficient stability across cases is crucial for reliable interpretation. Additionally, the choice of LASSO regularization parameter λ through experimental selection rather than data-driven optimization introduces subjectivity, though this was somewhat reduced by the subsequent OLS refinement stage.

Principally, the delay-coordinate representation of forced systems, while mathematically rigorous, produces models that are challenging to interpret in domains where principal knowledge exists. For the ECG application, the transformation from variables to abstract delay interactions obscures the relationship between identified terms and known physiological mechanisms. This interpretability gap currently limits the framework's clinical utility despite its impressive predictive capabilities.

6.3 Future Research Directions

Three key areas emerge as priorities for future research to address the current limitations. First, the development of adaptive algorithms for automatic parameter selection could substantially improve the framework's usability. Information-theoretic criteria, such as the Akaike Information Criterion (AIC) or Bayesian Optimization techniques, show promise for determining optimal embedding dimensions, delay times, and polynomial degrees directly from data. Results of the experiments with mutual information for delay selection and embedding dimension for estimation have yielded promising results worth advancing.

Second, integration of physically meaningful basis functions tailored to specific application domains could bridge the interpretability gap. For ECG analysis, incorporating potential models or equations as library terms might yield more physiologically transparent representations. Similarly, applications in various scientific disciplines could benefit from domain-specific features from pre-existing equations or behavior from existing models. This hybrid approach combining data-driven discovery with domain knowledge represents a promising middle ground between purely experimental and purely theoretical modeling.

Third, incorporation of Bayesian inference techniques would provide natural uncertainty quantification while maintaining the framework's sparse structure. Bayesian variants of LASSO

using hierarchical priors could simultaneously perform variable selection and provide posterior distributions over coefficients. Such probabilistic interpretations would be particularly valuable in applications where understanding model confidence is as important as the predictions themselves. Additional extensions could explore the use of Gaussian processes for derivative estimation and ordinary differential equations for handling highly nonlinear systems.

The successful application to ECG analysis suggests numerous opportunities in biomedical engineering. Future work should validate the framework on larger, labeled datasets encompassing diverse cardiac pathologies. Specific investigations can be made with the developed methodology to discover novel dynamical biomarkers for critical health conditions that may not be apparent in traditional time-domain analysis. More generally, the methodology's general mathematical foundation makes it applicable to other oscillatory systems in biology, engineering, and physics, provided appropriate adjustments are made to the function library and preprocessing steps for each domain.

6.4 Broader Implications

This research establishes that sparse identification methods, when properly regularized and validated, can indeed discover interpretable dynamical systems from data. The framework's success in both synthetic benchmarks and real-world ECG signals demonstrates its versatility across different levels of measurement noise and system complexity. However, the results also underscore that such approaches are not universal solutions and that their effectiveness depends critically on appropriate problem formulation, careful parameter selection, and domain-informed interpretation of results.

For the Van der Pol oscillator, the findings provide new insights into data-driven modeling of nonlinear damping phenomena. The consistent recovery of the essential terms

across noise levels and forcing conditions suggests that sparse regression can reliably identify core dynamical mechanisms even when features are distorted. This capability has immediate applications in engineering systems where fundamental models are incomplete but experimental data is abundant.

In clinical applications, while the current implementation may not replace traditional ECG analysis, it offers complementary capabilities for uncovering hidden dynamical patterns. The framework's ability to reconstruct phase space relationships from single-lead recordings could enable new approaches to arrhythmia characterization that go beyond conventional time-domain or frequency-domain features. This aligns with growing recognition in cardiology that nonlinear dynamics play crucial roles in cardiac pathophysiology but have been underutilized in clinical practice due to lack of practical analysis tools.

The methodological contributions extend beyond specific applications to general challenges in data-driven discovery. The systematic comparison of LASSO and OLS refinement strategies provides experimental evidence for hybrid approaches that combine the strengths of different regression techniques. Similarly, the investigation of delay embedding parameters offers practical guidelines for balancing reconstruction accuracy against model complexity in real-world settings. These insights will inform future developments in machine learning for dynamical systems, particularly as interest grows in scientific applications.

Ultimately, this work moves the field toward more reliable data-driven modeling by demonstrating that mathematical rigor and practical utility need not be competing priorities. The framework's careful attention to both theoretical foundations, through Takens' theorem and sparse regression theory, and implementation considerations, through noise-robust preprocessing and validation, provides a template for future research at the intersection of applied mathematics

and domain sciences. As measurement technologies continue advancing across disciplines, such rigorous yet flexible approaches will become increasingly vital for extracting meaningful knowledge from complex observational data.

CHAPTER 7

CONCLUSION

1.1 Summary of Contributions

This thesis has presented a comprehensive framework for data-driven identification of nonlinear dynamical systems, with rigorous validation across both synthetic benchmarks and real-world physiological signals. The methodology successfully bridges theoretical foundations from dynamical systems theory with practical machine learning tools, demonstrating that sparse regression techniques, when carefully structured and regularized, can extract interpretable governing equations from noisy observational data. The Van der Pol oscillator served as an ideal test case, revealing that the hybrid sparse regression approach recovers ground-truth equations with high coefficient accuracy in noiseless conditions while maintaining sufficient fidelity under substantial noise ($\sigma = 0.1$), outperforming solely using LASSO in term selection precision.

The study's insights into delay embedding for forced systems demonstrate broad implications for real-world applications where partial observations are common. While Takens' theorem provides theoretical guarantees, this work experimentally established that embedding dimensions $m = 1, 2, 3$ for the Van der Pol system strike a critical balance between capturing dynamical behavior and avoiding overfitting. This finding resonates beyond synthetic examples, as demonstrated in the ECG case study where $m = 7$ delay coordinates sufficiently reconstructed cardiac electrical activity. The framework's reliance on manual parameter tuning remains a limitation, but the systematic analysis provides clear pathways for future automation through information-theoretic criteria.

1.2 Implications for Theory and Practice

The research advances the field of system identification both methodologically and practically. On a theoretical level, it provides experimental validation of sparse regression techniques for nonlinear dynamical systems, particularly in challenging systems with noise and forcing effects. The consistent recovery of the Van der Pol oscillator's characteristic nonlinear damping behavior across various perturbations confirms that data-driven methods can indeed discover core physical mechanisms when properly constrained. The integration of Takens' Embedding Theorem with modern regression tools offers a principled approach to handling partial observations, extending the framework's applicability to real-world scenarios where complete state measurements are not always available.

For practical applications, the successful implementation on ECG signals demonstrates the framework's potential in biomedical engineering. While clinical interpretation of delay-coordinate models requires further development, the ability to reconstruct pathological cardiac dynamics from single-lead recordings opens new possibilities for arrhythmia characterization. The methodology's noise robustness suggests utility in scenarios where the quality of measurement data is often compromised. The approach can have possible applications in systems within scientific disciplines, where data-driven models can be utilized to complement or enhance the understanding or interpretation of dynamical systems.

1.3 Outlook

The work completed in this thesis suggests several promising directions for future research. The most immediate opportunity lies in automating parameter selection through adaptive algorithms that promote optimization techniques that combine information and theoretical criteria. Development of domain-specific function libraries represents another critical

direction since various disciplines may require incorporating domain-specific knowledge that can enhance interpretability. The integration of advanced derivative estimation methods, such as total variation regularization, may further improve performance on noisy experimental data.

The framework could be extended to handle non-autonomous systems with time-varying parameters or coupled oscillator networks that are common in systems within scientific disciplines. The developed mathematical foundation provides a basis for more complex scenarios. As measurement technologies continue advancing across scientific disciplines, data-driven system identification methods will become increasingly vital tools for extracting meaningful patterns from complex observational data. The careful balance between mathematical rigor and practical utility serves as a model for future developments in the field.

Ultimately, this research demonstrates that sparse, interpretable models can achieve competitive performance with more complex approaches while retaining physical plausibility. As scientific studies become increasingly data-rich, such methods that bridge theory and observations will play a crucial role in advancing our understanding of complex dynamical systems across physics, engineering, and biology. The framework developed here provides both specific tools for system identification and a general template for data-driven discovery that respects fundamental principles of mathematical modeling.

APPENDICES

APPENDIX A

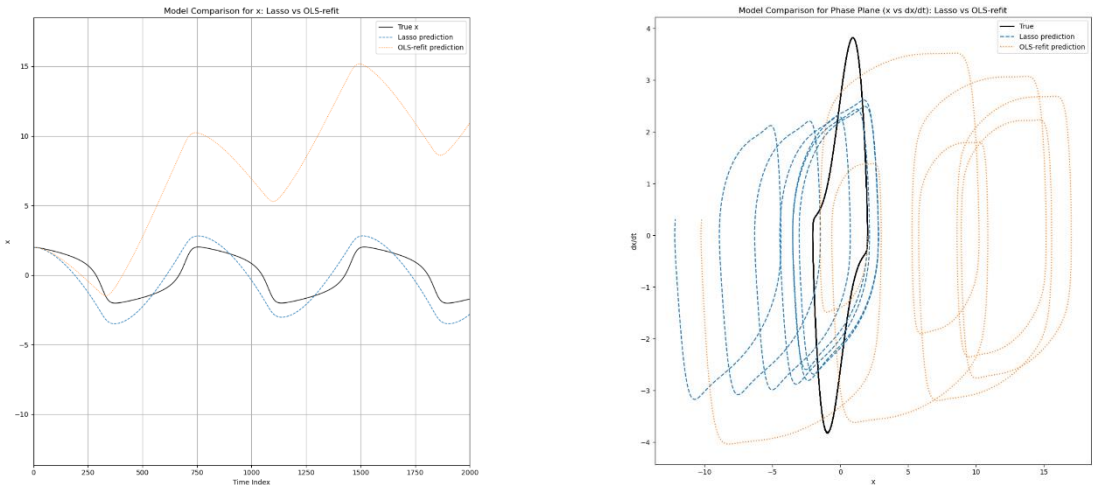
SECOND-ORDER VAN DER POL OSCILLATOR ANALYSIS

The second-order Van der Pol oscillator, expressed as

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

was carefully analyzed but ultimately not selected as the primary focus of this thesis due to fundamental numerical challenges in its data-driven identification. While mathematically equivalent to the first-order system studied in Chapters 3-4, the second-order formulation introduces compound errors that significantly degrade reconstruction accuracy. These limitations stem from the successive numerical operations required, where estimating the second derivative through finite differences amplifies measurement noise, while subsequent double integration steps accumulate truncation errors that grow quadratically over time.

Numerical experiments revealed these limitations through several observable effects. When applied to synthetic data generated from the second-order equation with $\mu = 2$ and $\Delta t = 0.01$, the LASSO-OLS framework failed to identify the characteristic nonlinear damping structure, as shown in Figure 31. The results are further confirmed through the state-space and phase plane approximations displayed in Figure 30 as the plots rapidly diverge from the ground truth systems. The phase portraits exhibited distorted limit cycles with high amplitude errors, contrasting with the first-order system's minimal errors under identical conditions.



(a) Position vs. Time (b) Position vs. Velocity

FIGURE 30. Baseline Second-Order Van der Pol Equation (Noiseless, Embedding Dimension $m = 1$).

```
# === Lasso-selected coefficients ===
('x_t_0', -25.568020)
('x_t_1', 24.705498)
('x_t_1*x_t_1', 0.079007)
('x_t_0*x_t_0*x_t_0', 35.346413)
('x_t_0*x_t_0*x_t_1', -11.103980)
('x_t_0*x_t_1*x_t_1', -12.101070)
('x_t_1*x_t_1*x_t_1', -12.185770)

# === OLS-refitted coefficients ===
('x_t_0', -0.957568)
('x_t_0*x_t_0*x_t_0', 32.848179)
('x_t_1*x_t_1*x_t_1', -32.863311)
```

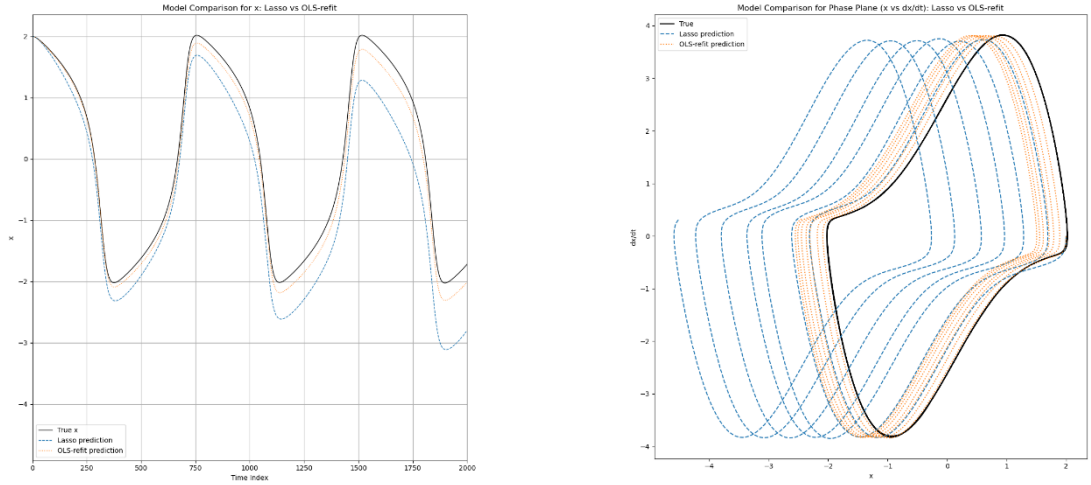
Terms and Coefficients for Modeling $\frac{d^2x}{dt^2}$

FIGURE 31. Feature Selection for Baseline Second-Order Van der Pol Equation (Noiseless, Embedding Dimension $m = 1$).

The function library for the second-order analysis maintained consistency with the first-order case, containing polynomial terms up to the third order $\left(x, \frac{dx}{dt}, x^2, x \cdot \frac{dx}{dt}, \left(\frac{dx}{dt}\right)^2, \ldots\right)$. However, the theoretical requirements for delay embedding dimensions were substantially higher, where Takens' theorem would suggest $m \geq 5$ for the implicit four-dimensional phase space rather than $m \geq 3$ for the first-order system. This increased dimensionality heightened the

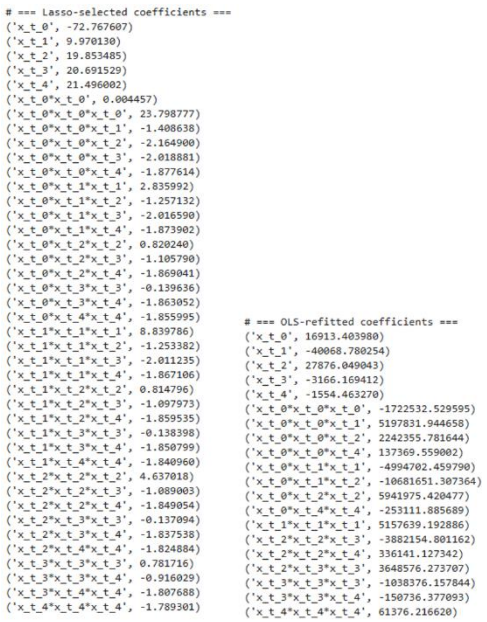
concern on dimensionality in sparse regression, requiring larger datasets and more computational resources for equivalent accuracy.

Application of Takens' embedding with dimension $m = 5$ improved the approximation of the hybrid regression method. Although the LASSO and OLS-refined models were able to capture the behavior of the dynamical system, both models displayed a drift of the plot over the set time interval, which was more evident in the LASSO model over the OLS-refined model and is displayed in Figure 32. When considering lower embedding dimension values for the second-order Van der Pol equation, the hybrid regression method failed to capture any resemblance of the true dynamics over time. As higher embedding dimensions lead to more terms being included in the function library, the selected terms and corresponding coefficients that display the resulting models with embedding dimension $m = 5$ used many terms while still failing to capture the true system dynamics, as evident in Figure 33. Therefore, for the second-order Van der Pol equation, increasing the embedding dimension would continue to obscure the possibility of recovering the true equation and dynamics.



(a) Position vs. Time (b) Position vs. Velocity

FIGURE 32. Baseline Second-Order Van der Pol Equation (Noiseless, Embedding Dimension $m = 5$).



Terms and Coefficients for Modeling $\frac{d^2x}{dt^2}$

FIGURE 33. Feature Selection for Baseline Second-Order Van der Pol Equation (Noiseless, Embedding Dimension $m = 5$).

This comprehensive analysis of the second-order Van der Pol oscillator provides important methodological insights for data-driven system identification. The results demonstrate that while second-order formulations are theoretically equivalent, their numerical implementation introduces substantial practical challenges that can outweigh their conceptual simplicity. The findings suggest that for many applied problems, reformulating higher-order systems as first-order equivalents may be essential for robust identification. Future work could explore hybrid approaches where second-order terms are identified from carefully preprocessed data or where physical constraints are incorporated to stabilize the regression, but the current results establish clear limitations of direct implementations.

The choice of mathematical formulation in data-driven modeling requires careful consideration of both theoretical completeness and practical numerical stability. For the Van der Pol oscillator and similar systems, the first-order formulation can provide superior performance

despite its increased variable count, offering an important consideration for applying these methods to more complex dynamical systems in engineering and scientific applications. These highlights contribute to the growing literature on robust system identification by highlighting how algorithmic choices interact with fundamental numerical analysis principles

APPENDIX B

PYTHON CODE FOR RESULTS IN CHAPTERS 4

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from sklearn.linear_model import Lasso, LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.feature_selection import SequentialFeatureSelector
from itertools import combinations_with_replacement

# 1. Simulate the Van der Pol system
def van_der_pol_forcing(t, y, mu, C, D):
    x, v = y
    dxdt = v
    dvdt = mu * (1 - x**2) * v - x + C * t + D * np.sin(t)
    return [dxdt, dvdt]

mu, C, D = 2, 0, 0
t_eval = np.linspace(0, 50, 5001)
sol = solve_ivp(van_der_pol_forcing, (0,50), [2.0, 0.0], args=(mu, C, D), t_eval=t_eval)
time, x, v = sol.t, sol.y[0], sol.y[1]

# 2. Compute true derivatives
dxdt_true = v
dvdt_true = mu * (1 - x**2) * v - x + C * time + D * np.sin(time)
```

```
# 3. Calculate approximate derivatives using forward Euler method
dt = time[1] - time[0]
dxdt_euler = np.diff(x)/dt
dvdt_euler = np.diff(v)/dt

# Add noise to approximated derivatives
np.random.seed(0)
dxdt_euler += np.random.normal(0, 0.1, size=dxdt_euler.shape)
dvdt_euler += np.random.normal(0, 0.1, size=dvdt_euler.shape)

# 4. Multivariate Takens' embedding of both x and v
def create_multivariate_embedding(x, v, tau, d):
    N = len(x) - (d - 1) * tau
    emb = np.zeros((N, 2*d))  # 2 variables (x and v) each with d delays
    for i in range(d):
        emb[:, 2*i]   = x[i*tau : i*tau + N]  # x delays
        emb[:, 2*i+1] = v[i*tau : i*tau + N]  # v delays
    return emb
tau = 1
embedding_dim = 1
degree = 3
threshold = 1e-5
```

```
embedding = create_multivariate_embedding(x, v, tau, embedding_dim)
embedding = embedding[:-1]
dxdt_target = dxdt_euler[:len(embedding)]
dvdt_target = dvdt_euler[:len(embedding)]
x_true = x[:len(embedding)]
v_true = v[:len(embedding)]

# 5. Build polynomial feature matrix
poly = PolynomialFeatures(degree=degree, include_bias=True)
Theta = poly.fit_transform(embedding)

# 6. Generate input symbol names dynamically
input_symbols = []
for i in range(embedding_dim):
    input_symbols.append(f'x_t_{i*tau}')
    input_symbols.append(f'v_t_{i*tau}')
feature_names = ['1'] + [
    "*".join(comb)
    for deg in range(1, degree + 1)
    for comb in combinations_with_replacement(input_symbols, deg)
]

# 7. Function to perform modeling process
```

```
def model_derivative(Theta, target, feature_names, threshold, derivative_name):
    print(f"\n\n==== Modeling {derivative_name} ====")
    # Fit Lasso regression to recover a sparse model
    lasso = Lasso(alpha=0.001, max_iter=10000).fit(Theta, target)
    # Display Lasso-selected features
    print("\n# === Lasso-selected coefficients ===")
    selected_idx = []
    for idx, coef in enumerate(lasso.coef_):
        if abs(coef) > threshold:
            print(f"('{feature_names[idx]}', {coef:.6f})")
            selected_idx.append(idx)
    # OLS + forward selection on Lasso-selected features
    def fit_ols_select(X, y, idxs):
        if len(idxs) <= 1:
            model = LinearRegression().fit(X[:, idxs], y)
            return model, np.array(idxs)
        sfs = SequentialFeatureSelector(
            LinearRegression(), direction='forward', scoring='r2', cv=5
        )
        sfs.fit(X[:, idxs], y)
        final_idxs = np.array(idxs)[sfs.get_support()]
        model = LinearRegression().fit(X[:, final_idxs], y)
        return model, final_idxs
```

```
    ols_model, final_idx = fit_ols_select(Theta, target, selected_idx)
    # Display OLS-refitted coefficients
    print("\n# === OLS-refitted coefficients ===")
    for i, idx in enumerate(final_idx):
        print(f"('{feature_names[idx]}', {ols_model.coef_[i]:.6f})")
    return lasso, ols_model, final_idx
# Model derivatives
lasso_dxdt, ols_dxdt, final_idx_dxdt = model_derivative(Theta, dxdt_target, feature_names,
threshold, 'dx/dt')
lasso_dvdt, ols_dvdt, final_idx_dvdt = model_derivative(Theta, dvdt_target, feature_names,
threshold, 'dv/dt')
# Compare Lasso vs OLS predictions
lasso_dxdt_pred = lasso_dxdt.predict(Theta)
ols_dxdt_pred = ols_dxdt.predict(Theta[:, final_idx_dxdt])
lasso_dvdt_pred = lasso_dvdt.predict(Theta)
ols_dvdt_pred = ols_dvdt.predict(Theta[:, final_idx_dvdt])


# 8. State prediction using current states
def predict_states(Theta, dxdt_model, dvdt_model, dxdt_idx, dvdt_idx, x0, v0, dt):
    N = len(Theta)
    x_pred = np.zeros(N+1)
    v_pred = np.zeros(N+1)
    x_pred[0], v_pred[0] = x0, v0
```

```
    # Initialize first point using the first embedded values
    x_pred[1], v_pred[1] = embedding[0,0], embedding[0,1]
    for i in range(1, N):
        # Get current features
        if isinstance(dxdt_model, LinearRegression):
            dxdt = dxdt_model.predict(Theta[i-1:i, dxdt_idx])[0]
        else:
            dxdt = dxdt_model.predict(Theta[i-1:i])[0]
        if isinstance(dvdt_model, LinearRegression):
            dvdt = dvdt_model.predict(Theta[i-1:i, dvdt_idx])[0]
        else:
            dvdt = dvdt_model.predict(Theta[i-1:i])[0]
        # Update states using forward Euler
        x_pred[i+1] = x_pred[i] + dxdt * dt
        v_pred[i+1] = v_pred[i] + dvdt * dt
    return x_pred[1:], v_pred[1:]  # Trim initial condition
# Calculate state predictions
x_pred_lasso, v_pred_lasso = predict_states(
    Theta, lasso_dxdt, lasso_dvdt, final_idx_dxdt, final_idx_dvdt, x[0], v[0], dt
)
x_pred_ols, v_pred_ols = predict_states(
    Theta, ols_dxdt, ols_dvdt, final_idx_dxdt, final_idx_dvdt, x[0], v[0], dt
)
```

```
# 8. Create all plots
plt.figure(figsize=(10, 50))
# Derivatives plots
plt.subplot(5, 1, 1)
plt.plot(dxdt_target, label='True dx/dt', color='black', lw=1)
plt.plot(lasso_dxdt_pred, label='Lasso prediction', linestyle='--', lw=1)
plt.plot(ols_dxdt_pred, label='OLS-refit prediction', linestyle=':', lw=1)
plt.xlabel('Time Index')
plt.ylabel('dx/dt')
plt.xlim(0, 2000)
plt.title('Model Comparison for dx/dt: Lasso vs OLS-refit')
plt.legend()
plt.grid()

plt.subplot(5, 1, 2)
plt.plot(dvdt_target, label='True dv/dt', color='black', lw=1)
plt.plot(lasso_dvdt_pred, label='Lasso prediction', linestyle='--', lw=1)
plt.plot(ols_dvdt_pred, label='OLS-refit prediction', linestyle=':', lw=1)
plt.xlabel('Time Index')
plt.ylabel('dv/dt')
plt.xlim(0, 2000)
plt.title('Model Comparison for dv/dt: Lasso vs OLS-refit')
```

```
plt.legend()
plt.grid()

# State variable plots
plt.subplot(5, 1, 3)
plt.plot(x_true, label='True x', color='black', lw=1)
plt.plot(x_pred_lasso, label='Lasso prediction', linestyle='--', lw=1)
plt.plot(x_pred_ols, label='OLS-refit prediction', linestyle=':', lw=1)
plt.xlabel('Time Index')
plt.ylabel('x')
plt.xlim(0, 2000)
plt.title('Model Comparison for x: Lasso vs OLS-refit')
plt.legend()
plt.grid()

plt.subplot(5, 1, 4)
plt.plot(v_true, label='True v', color='black', lw=1)
plt.plot(v_pred_lasso, label='Lasso prediction', linestyle='--', lw=1)
plt.plot(v_pred_ols, label='OLS-refit prediction', linestyle=':', lw=1)
plt.xlabel('Time Index')
plt.ylabel('v')
plt.xlim(0, 2000)
plt.title('Model Comparison for v: Lasso vs OLS-refit')
```

```
plt.legend()
plt.grid()

# Phase space plots
plt.subplot(5, 1, 5)
plt.plot(x_true, v_true, color='black', label='True')
plt.plot(x_pred_lasso, v_pred_lasso, label='Lasso prediction', linestyle='--')
plt.plot(x_pred_ols, v_pred_ols, label='OLS-refit prediction', linestyle=':')
plt.xlabel('x')
plt.ylabel('v')
plt.title('Model Comparison for Phase Plane: Lasso vs OLS-refit')
plt.legend()

plt.tight_layout()
plt.show()
```

REFERENCES

REFERENCES

Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems." Proceedings of the national academy of sciences 113, no. 15 (2016): 3932-3937.

Corbetta, Matteo. "Application of sparse identification of nonlinear dynamics for physics-informed learning." In *2020 IEEE Aerospace Conference*, pp. 1-8. IEEE, 2020.

Goyal, Pawan, and Peter Benner. "Discovery of nonlinear dynamical systems using a Runge–Kutta inspired dictionary-based sparse regression approach." *Proceedings of the Royal Society A* 478, no. 2262 (2022): 20210883.

Kiser, Shawn L., Mikhail Guskov, Marc Rébillat, and Nicolas Ranc. "Exact identification of nonlinear dynamical systems by Trimmed Lasso." *arXiv preprint arXiv:2308.01891* (2023).

Kovacic, Ivana, and Ronald E. Mickens. "A generalized van der Pol type oscillator: Investigation of the properties of its limit cycle." Mathematical and Computer Modelling 55.3-4 (2012): 645-653.

Yap, Han Lun, and Christopher J. Rozell. "Stable takens' embeddings for linear dynamical systems." IEEE transactions on signal processing 59.10 (2011): 4781-4794.