Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI DI UDINE
HIC SUNT FUTURA

Corso di dottorato di ricerca in:

" Mathematical and Physical Sciences "

Università degli Studi di Udine

Ciclo 38°

Titolo della tesi

"Sparse Identification of Delay Equations and Structured Populations"

Dottorando
Muhammad Tanveer

Supervisore
Dimitri Breda

Co-supervisore
Rossana Vermiglio

Anno 2025

Finanziato dall'Unione Europea - Next Generation EU, Missione 4 Componente 1 CUP G23C22001320003
(Codice corso: DOT215B2RJ) (D.M. 351/2022)

UNIVERSITY OF UDINE

DEPARTMENT OF MATHEMATICS, COMPUTER SCIENCE AND PHYSICS

PHD IN MATHEMATICAL AND PHYSICAL SCIENCES
CYCLE XXXVIII

PH.D. THESIS

Sparse Identification of Delay Equations and Structured Populations

CANDIDATE:
Muhammad Tanveer

SUPERVISOR:
Dimitri Breda

CO-SUPERVISOR:
Rossana Vermiglio

Academic Year 2024/2025

Muhammad Tanveer

Sparse Identification of Delay Equations and Structured Populations

---

Author's e-mail: mtanveer.161994@gmail.com

Author's address:

Dipartimento di Matematica e Informatica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

This document was written with LaTeX using the Marco Comini's (University of Udine) Packages/Classes https://users.dimi.uniud.it/~marco.comini/LaTeX/.

Dedicated to my parents and all my family members.

يُؤْتِى ٱلْحِكْمَةَ مَن يَشَآءُ ۚ وَمَن يُؤْتَ ٱلْحِكْمَةَ فَقَدْ أُوتِىَ خَيْرًا كَثِيرًا ۗ وَمَا يَذَّكَّرُ إِلَّآ أُو۟لُوا۟ ٱلْأَلْبَـٰبِ

"He grants wisdom to whoever He wills. And whoever is granted wisdom is certainly blessed with a great privilege. But none will be mindful "of this" except people of reason. "

"Al Quran", (Sūrat al-Bāqārāh, No 2, Āyat 269).

Abstract

Dynamical systems exhibiting memory effects, where current evolution depends on past states, present fundamental challenges for data-driven equation discovery. This thesis develops complementary methodologies for discovering governing equations of such systems from time-series data by extending the Sparse Identification of Nonlinear Dynamics framework to three principal classes of delayed dynamical systems: deterministic delay differential equations with discrete delays, stochastic delay differential equations, and distributed delay systems.

For delay differential equations with discrete delays, we introduce two approaches with distinct trade-offs. Expert-SINDy combines sparse regression with external particle swarm optimisation to simultaneously discover functional forms and discrete delay parameters when the system structure is partially known, maintaining interpretability of the resulting equations. Pragmatic-SINDy reformulates the infinite-dimensional identification problem using pseudospectral collocation, reducing it to finite-dimensional optimisation over only the maximum delay, thereby achieving improved computational efficiency at the cost of reduced equation interpretability. Comprehensive validation on logistic delay equations, Mackey-Glass dynamics, and neural network models demonstrates robust delay parameter identification and accurate trajectory reconstruction across diverse nonlinear behaviours.

We then extend the framework to stochastic delay differential equations, developing Itô-Taylor expansion-based moment estimators that reconstruct drift and diffusion components from noisy observations. Three practical estimation strategies accommodate different data availability scenarios, with extensive computational analysis establishing that pre-regression averaging substantially outperforms alternatives.

For systems with distributed memory, we develop quadrature-based kernel identification methods that recover nonautonomous delay kernels without requiring prior functional form specification. Joint optimisation of integration bounds and kernel parameters achieves systematic discovery from data alone. Applications to logistic renewal equations and structured population dynamics validate the capability of the framework to extract interpretable sparse representations of distributed delay systems.

Practical utility is demonstrated through application to Severe Fever with Thrombocytopenia Syndrome transmission dynamics, where distributed delay models are discovered directly from real surveillance data. Finally, comparative analysis contextualises sparse identification methods within the broader landscape of data-driven system identification, examining trade-offs between interpretability, computational efficiency, and data requirements compared with neural network approaches.

MATLAB implementations of the methods presented in this thesis are provided at http://cdlab.uniud.it/software.

Sommario

I sistemi dinamici che presentano effetti di memoria, in cui l'evoluzione attuale dipende dagli stati passati, pongono sfide fondamentali per la scoperta di equazioni basate sui dati. Questa tesi sviluppa metodologie complementari per scoprire le equazioni che governano tali sistemi a partire da serie di dati temporali, estendendo il quadro dell identificazione sparsa delle dinamiche non lineari a tre classi principali di sistemi dinamici ritardati: equazioni differenziali deterministiche con ritardo discreto, equazioni differenziali stocastiche con ritardo e sistemi con ritardo distribuito.

Per le equazioni differenziali con ritardo discreto, introduciamo due approcci con caratteristiche diverse. Expert-SINDy combina la regressione sparsa con l'ottimizzazione esterna particle swarm per scoprire simultaneamente la forma funzionale e i parametri di ritardo discreti quando la struttura del sistema è parzialmente nota, mantenendo l'interpretabilità delle equazioni risultanti. Pragmatic-SINDy riformula il problema dell'identificazione infinito-dimensionale utilizzando la collocazione pseudospettrale, riducendolo ad un'ottimizzazione finito-dimensionale solo sul ritardo massimo, ottenendo così una maggiore efficienza computazionale a scapito di una ridotta interpretabilità dell'equazione. Una validazione completa su equazioni di ritardo logistico, dinamiche Mackey-Glass e modelli di reti neurali dimostra una robusta identificazione dei parametri di ritardo e un'accurata ricostruzione della traiettoria in diversi comportamenti non lineari.

Estendiamo quindi il quadro alle equazioni differenziali stocastiche con ritardo, sviluppando stimatori di momento basati sull'espansione di Itô-Taylor che ricostruiscono le componenti di drift e diffusione da osservazioni soggette a rumore. Tre strategie di stima pratiche si adattano a diversi scenari di disponibilità dei dati, con un'analisi computazionale approfondita che stabilisce che la media pre-regressione supera sostanzialmente le alternative.

Per i sistemi con memoria distribuita, sviluppiamo metodi di identificazione del kernel basati sulla quadratura che recuperano i kernel di ritardo non autonomi senza richiedere una specifica preliminare della forma funzionale. L'ottimizzazione congiunta degli estremi di integrazione e dei parametri del kernel consente una scoperta sistematica basata esclusivamente sui dati. Le applicazioni alle equazioni di rinnovo logistico e alle dinamiche di popolazione strutturate convalidano la capacità del framework di estrarre rappresentazioni sparse interpretabili dei sistemi di ritardo distribuiti.

L'utilità pratica è dimostrata attraverso l'applicazione alle dinamiche di trasmissione della sindrome da febbre grave con trombocitopenia, dove i modelli di ritardo distribuito vengono scoperti direttamente dai dati reali di sorveglianza. Infine, l'analisi comparativa contestualizza i metodi di identificazione sparsa nel panorama più ampio dell'identificazione dei sistemi basata sui dati, esaminando i compromessi tra interpretabilità, efficienza computazionale e requisiti dei dati rispetto agli approcci basati sulle reti neurali.

Le implementazioni MATLAB dei metodi presentati in questa tesi sono disponibili all'indirizzo `http://cdlab.uniud.it/software`.

Acknowledgments

First and foremost, all praise and gratitude are due to Allah (SWT), the Most Gracious and the Most Merciful, for bestowing upon me the strength, patience, and knowledge to complete this journey.

I would like to express my deepest gratitude to my supervisor, Prof. Dimitri Breda, for his invaluable guidance, continuous encouragement, and the vast knowledge he shared with me throughout this research. His insights were fundamental to the development of the methods presented in this thesis. I am also extremely grateful to my co-supervisor, Prof. Rossana Vermiglio, for her support, and expert advice during my doctoral studies.

I would like to extend my gratitude to Prof. Gábor Orosz and Dr. Xunbi A. Ji for their invaluable collaboration on the book chapter we co-authored. Their insights into data-driven methods for delay differential equations significantly improved the quality of that joint work.

My sincere thanks go to the Department of Mathematics, Computer Science and Physics at the University of Udine and the CDLab for providing a stimulating research environment. I also acknowledge the financial support provided by the Italian Ministry of University and Research (MUR) through the PhD grant PNRR DM351/2022 and DM352/2022 (CUP: G23C22001320003).

I would also like to express my heartfelt thanks to a dear friend at the University of Udine for the constant support, encouragement, and the memorable moments we shared during this journey.

Finally, I am deeply grateful to my parents and my family. Their unconditional love, prayers, and sacrifices have been my greatest strength. A special hug goes to my beloved nephews and niece, *Muhammad Zarrar*, *Muhammad Saleh* and *Wania Zaheer*, for bringing so much joy into my life.

Contents

Abbreviations

- BB-SINDy - Blackbox SINDy
- BF - Brute-Force Optimisation
- BO - Bayesian Optimisation
- CD - Central Difference
- DD-SINDy - Distributed Delay SINDy
- DDE - Delay Differential Equation
- DIDE - Delay Integro-Differential Equation
- E-SINDy - Expert SINDy
- FD - Forward Difference
- ISINDy - Integral SINDy
- IVP - Initial Value Problem
- KM - Kramers-Moyal
- LASSO - Least Absolute Shrinkage and Selection Operator
- NDDE - Neural Delay Differential Equation
- NODE - Neural Ordinary Differential Equation
- ODE - Ordinary Differential Equation
- P-SINDy - Pragmatic SINDy
- PDE - Partial Differential Equation
- PS - Particle Swarm Optimisation
- RE - Renewal Equation
- RHS - Right-hand side
- RMSE - Root Mean Square Error
- SDDE - Stochastic Delay Differential Equation
- SFTS - Severe Fever with Thrombocytopenia Syndrome
- SFTSV - Severe Fever with Thrombocytopenia Syndrome Virus
- SINDy - Sparse Identification of Nonlinear Dynamics
- SODE/SDE - Stochastic Ordinary Differential Equation

- STLS - Sequential Thresholded Least Squares
- TR - Trapezoidal Rule

Throughout this thesis, the term SINDy refers to the standard SINDy framework, unless otherwise specified.

Introduction

Traditionally, mathematical modelling has depended on first-principles methods utilising profound theoretical understanding to derive governing equations. While effective in physical and engineering sciences, this approach encounters significant challenges when applied, e.g., to systems characterised by intricate interactions, high dimensional nonlinearities, and inherent memory effects. Creating accurate mechanistic models becomes exceedingly difficult, especially when fundamental physical laws are not fully understood or when systems display temporal dependencies that cannot be captured by instantaneous relationships alone.

The emergence of large-scale data collection and advanced computational resources has sparked a shift toward data-driven modelling approaches. These methods derive governing equations directly from observational data, bypassing the need for comprehensive theoretical derivation. Sparse Identification of Nonlinear Dynamics (SINDy) [31] has emerged as a particularly effective framework that balances model simplicity, interpretability, and predictive accuracy. By employing sparsity-promoting regression techniques, SINDy identifies minimal sets of active terms from function libraries, producing interpretable governing equations that reveal fundamental system mechanisms.

Although SINDy has achieved significant success with ordinary differential equations (ODEs), numerous real-world systems display memory effects where the current state evolution depends not only on present conditions but also on the system's past. These temporal dependencies manifest in physiological processes with signal propagation delays [5, 69], neural activity with synaptic transmission delays [120, 129], epidemiological dynamics with incubation periods, and population dynamics with maturation stages [130]. Such memory-dependent dynamics are best described by delay differential equations (DDEs) [47, 62], which extend ODEs by allowing the rate of change to depend on past states at specific time delays. When memory effects are distributed continuously over intervals representing age structure or maturation periods, the suitable mathematical framework involves delay integro-differential equations (DIDEs) and renewal equations (REs) with distributed delay kernels [15].

Extending data-driven identification to systems with delays introduces fundamental challenges that distinguish this problem from the well-explored ODE case. First, unknown delay parameters transform identification from convex sparse regression into nonconvex global optimisation over both functional form and delay values. Second, the infinite-dimensional state space inherent in delay systems adds computational complexity that scales poorly with standard methods. Third, distributed delays involve nonautonomous kernel functions depending on both the integration variable and state history, requiring specialised identification techniques. Fourth, simultaneous stochastic fluctuations and delay effects create non-Markovian dynamics that challenge standard moment estimation techniques. These difficulties have limited data-driven delay system identification to situations with substantial prior knowledge or computationally prohibitive exhaustive searches [104, 116].

This thesis systematically extends the SINDy framework to delayed dynamical systems across multiple complexity levels: deterministic DDEs with discrete delays, stochastic delay differential equations (SDDEs) combining randomness with memory effects, and systems with distributed delays represented by integral kernels. The developed methodologies enable automatic discovery of governing equations and delay parameters directly from time-series data, significantly reducing reliance on prior mechanistic knowledge while maintaining the interpretability benefits distinguishing sparse identification from black-box machine learning approaches.

I.1 Problem statement

This thesis addresses the data-driven identification of governing equations for dynamical systems exhibiting memory effects through temporal delays. Formally, consider a dynamical system whose state evolution depends on its history over some temporal window. The challenge is to discover both the functional form of the governing equations and the characteristic delay parameters from discrete time-series measurements $\{x(t_i)\}_{i=1}^{m}$, where $x(t) \in \mathbb{R}^n$ represents the system state variables and measurements may be corrupted by noise or limited in quantity.

For deterministic systems with discrete delays, the governing equations take the form

$$x'(t) = f(x(t), x(t-\tau_1), \ldots, x(t-\tau_k)),$$

where $f : \mathbb{R}^{n(k+1)} \to \mathbb{R}^n$ represents the unknown right-hand side and $0 < \tau_1 < \cdots < \tau_k$ denotes possibly unknown delay parameters. Traditional system identification approaches assume either complete knowledge of the delay values or restrict attention to scenarios where delays can be inferred through separate physical reasoning. This thesis removes these restrictive assumptions, developing methods that simultaneously identify both f and $\{\tau_j\}_{j=1}^{k}$ from data.

When stochastic fluctuations coexist with delay effects, the system evolves according to SDDEs of the form

$$\mathrm{d}X(t) = f(X(t), X(t-\tau))\ \mathrm{d}t + g(X(t), X(t-\tau))\ \mathrm{d}W(t),$$

where $W(t)$ denotes a Wiener process, f governs the deterministic drift, and g controls the diffusion intensity. The identification challenge now requires simultaneously recovering both drift and diffusion functions from noisy realisations, a problem substantially more difficult than the deterministic case due to the amplification of measurement noise through moment ratio computations.

For systems exhibiting distributed memory effects, where evolution depends on a weighted history over continuous intervals rather than discrete past points, the governing equations assume the form

$$x'(t) = \int_{-\tau}^{0} \gamma(\sigma, x(t+\sigma))\, d\sigma$$

for DIDEs, or

$$x(t) = \int_{-\tau}^{0} \gamma(\sigma, x(t+\sigma))\, d\sigma$$

for REs. Here, the kernel function $\gamma : [-\tau, 0] \times \mathbb{R}^n \to \mathbb{R}^n$ is nonautonomous, depending explicitly on both the delay variable σ and the state history. Identifying such kernels from data presents unique challenges: the functional space is infinite-dimensional, the integration bounds may be unknown.

The specific general challenges addressed across these problem classes include:

- When multiple delays are unknown, an exhaustive search over delay parameter spaces becomes computationally prohibitive, scaling exponentially with the number of delays.
- Sparse regression performance depends critically on including appropriate candidate functions in the library; missing key nonlinear terms leads to spurious fitted models.
- Numerical differentiation and moment estimation amplify measurement noise, particularly degrading diffusion term identification in stochastic systems.
- Multiple sparse models may fit available data equally well when information is insufficient, creating fundamental identifiability limitations.
- Distributed delay kernels depend on both integration variables and state history, requiring specialised library construction techniques beyond standard polynomial or trigonometric bases.

I.2 State of the art

Data-driven system identification has emerged as a powerful paradigm for extracting governing equations from observational data. A broad spectrum of approaches has been developed, ranging from black-box machine learning techniques such as neural ordinary and delay differential equations [39, 61, 71, 73, 74, 89], physics-informed neural networks [108], and Gaussian process regression [110] and spectral methods including dynamic mode decomposition [118] and Koopman operator regression [83]. While these methods can achieve high predictive accuracy, they often prioritise forecasting performance over interpretability and do not necessarily yield explicit low-dimensional governing equations. The SINDy framework [31] represents a breakthrough by using sparsity-promoting regression to identify parsimonious models from data. Originally developed for ODEs [31], SINDy has been extended to PDEs [115], SDEs [12, 139], and more recently to DDEs with discrete delays [16, 84, 104, 116]. This contextualises sparse identification within the broader landscape of data-driven system identification.

DDEs with discrete delays have been extensively studied from theoretical perspectives [47, 62, 122]. Classical approaches to DDE identification typically assume prior knowledge of delay values or employ computationally expensive grid searches [51, 141]. The method proposed by [116], here referred to as Expert SINDy (E-SINDy), extends SINDy to include delayed variables when the number and values of delays are known a priori. When this information is missing, brute-force (BF) optimisation over candidate delay values becomes necessary, creating computational bottlenecks. Recent advances incorporate Bayesian optimisation (BO) strategies [104] to guide the search over delay parameter space, offering improved computational efficiency. However, the computational cost still scales poorly with the number of unknown delays, motivating the development of more efficient reformulations.

SDDEs provide natural frameworks for modelling systems subject to both random fluctuations and memory effects. The theoretical foundations are well-established [32, 66, 82, 101, 102], and the robustness of SDDEs has been rigorously analysed [95]. Data-driven identification of stochastic systems without delays has advanced considerably through extensions of SINDy employing Kramers-Moyal moment estimators [12, 139]. However, extension to SDDEs remains largely unexplored. The recent work by [64] restricts attention to the Stratonovich formulation and relies on small-delay Taylor approximations, limiting generalisation to arbitrary delays and multiple delay structures.

Distributed delay systems have been extensively studied from theoretical perspectives, and the integral SINDy (ISINDy) framework [96, 117, 140] provides a foundation by replacing numerical differentiation with integration. This approach shows promise for handling integral functionals because integration is inherently more stable than differentiation in the presence of measurement noise. However, existing methods do not address the nonautonomous nature of distributed kernels, requiring substantial methodological innovation for data-driven discovery.

The literature review is consolidated, drawing from Section I.2 above and complemented by detailed domain-specific discussions in each chapter: Chapter 2, Section 2.1.2 (SINDy for DDEs); Chapter 3, Section 3.1.2 (SINDy for SDDEs); and Chapter 4, Section 4.1.2 (distributed delays and structured populations).

I.3 Gaps in existing methods

Despite considerable progress in data-driven system identification and theoretical analysis of delay systems, critical gaps remain:

- most methods require a priori knowledge of delay values, functional forms, or kernel structures;
- computational cost scales poorly with the number of potential delays;
- simultaneous treatment of stochastic fluctuations and delay effects remains largely unexplored;
- data-driven discovery of nonautonomous kernels lacks systematic approaches maintaining interpretability;
- trade-offs between accuracy, computational tractability, and sparsity are not well characterised.

These gaps motivate the development of new approaches and analysis that can discover governing equations and delay parameters efficiently while maintaining the sparsity and interpretability advantages of the SINDy framework.

In this thesis, measurement noise is not treated explicitly. All numerical experiments use either noise-free data or data with only process noise, and derivative estimates are computed under this assumption. Extending the framework to account for observation noise in a systematic way for example using integral SINDy, suitable denoising procedures, or explicit statistical noise models remains an important direction for future work.

I.4 Computational framework

All numerical experiments presented in this thesis were run on a Windows 11 system (CPU 5 GHz, RAM 16 GB) using MATLAB implementations for SINDy (available at `http://cdlab.uniud.it/software`). The computational implementations leverage MATLAB version (R2024a). All numerical experiments in this thesis utilise integrated MATLAB optimisers from standard toolboxes:

- `bayesopt.m` for Bayesian Optimisation (BO) from the *Statistics and Machine Learning Toolbox*
- `particleswarm.m` for Particle Swarm (PS) from the *Global Optimisation Toolbox*
- STLS (Sequentially Thresholded Least Squares) and LASSO implementations for sparse regression

The developed methods are implemented through MATLAB scripts designed for reproducibility and ease of use. For E-SINDy:

(*i*) open `ESINDy_main.m`

(*ii*) modify the `User inputs` section according to the desired model, time window, sample distribution, and noise amplitude

(*iii*) execute `ESINDy_main` from the Command Window

Similarly, for P-SINDy:

(*i*) open `PSINDy_main.m`

(*ii*) adjust user parameters (particularly the collocation degree M) in the `User inputs` section

(*iii*) execute `PSINDy_main` from the Command Window

Each script automatically generates data for the selected model, constructs the library of candidate functions, optionally applies the chosen optimiser to estimate unknown delays and parameters and finally carries out the sparse regression step (via STLS).

I.5 Methodology

This thesis extends the SINDy framework to delayed dynamical systems across multiple classes of complexity.

I.5.1 Expert SINDy

E-SINDy extends the classical SINDy framework to DDEs when delay values are known or can be estimated. The method incorporates delayed copies of state variables into the library matrix and applies standard sparse regression. External optimisation (PS [16], BO, or BF) can search over delay parameter spaces when delays are unknown. See Chapter 2 for detailed development and analysis.

I.5.2 Pragmatic SINDy

P-SINDy reformulates the infinite-dimensional DDE identification problem using pseudospectral collocation on Chebyshev nodes, reducing it to a finite-dimensional ODE system [22, 24]. This approach requires optimisation of only the maximum delay value, offering improved computational efficiency at the cost of reduced interpretability. See Chapter 2 for detailed development and analysis.

I.5.3 Extension to stochastic delay differential equations.

Chapter 3 extends SINDy to SDDEs by employing Itô-Taylor expansion-based estimators (Kramers-Moyal, forward/central differences, trapezoidal rules) for reconstructing drift and diffusion from time-series data. Three practical approaches accommodate different data availability scenarios and statistical robustness requirements. Extensive computational analysis establishes that pre-regression averaging substantially outperforms alternatives.

I.5.4 Distributed delay kernel identification.

Chapter 4 develops methods for identifying DIDEs and REs with distributed memory kernels. Quadrature-based kernel libraries explicitly include both delay variables and state history, enabling recovery of nonautonomous kernels. Extended sparse regression identifies analytical kernel structure through weighted sums of quadrature terms. Joint optimisation of integration bounds and kernel parameters achieves systematic discovery without prior knowledge.

I.5.5 Comparative analysis with neural network approaches.

Chapter 6 provides comparative analysis of E-SINDy, P-SINDy, and neural delay differential equation methodologies across interpretability, delay parameter estimation, computational efficiency, and data requirements.

I.6 Objectives

The principal objectives of this thesis are:

(*i*) to develop E-SINDy with efficient external optimisation for unknown delays in deterministic DDEs;

(*ii*) to introduce P-SINDy based on pseudospectral collocation for improved computational efficiency;

(*iii*) to extend SINDy to SDDEs with comprehensive treatment of drift and diffusion estimation;

(*iv*) to develop methods for discovering distributed delay kernels from data without requiring prior functional form specification;

(*v*) to provide comprehensive comparison of sparse identification approaches with neural network methods;

(*vi*) to validate all methodologies through numerical experiments on benchmark and real world SFTS (Severe Fever with Thrombocytopenia Syndrome).

I.7 Results

The thesis develops complementary methodologies for data-driven identification across a wide spectrum of delayed dynamical systems:

Chapter 2: *SINDy for Delay Differential Equations* establishes E-SINDy and P-SINDy methods for deterministic DDEs with discrete delay. E-SINDy combined with PS provides robust delay parameter identification with interpretability. P-SINDy trades some interpretability for improved computational efficiency by requiring optimisation of only the maximum delay. Numerical validation on logistic delay equations and Rössler systems demonstrates the effectiveness and trade-offs of both approaches.

Chapter 3: *SINDy for Stochastic Delay Differential Equations* develops Itô-Taylor expansion based estimators for reconstructing drift and diffusion from time-series data. Three practical approaches accommodate different data availability scenarios. Extensive computational analysis demonstrates that pre-regression averaging (Approach B1) substantially outperforms alternatives across multiple metrics.

Chapter 4: *SINDy for Structured Populations with Distributed Memory Effects* constructs quadrature-based kernel libraries and develops extended sparse regression for identifying nonautonomous kernels. Joint optimisation of integration bounds and kernel parameters achieves automatic discovery without prior knowledge. Comprehensive experiments on logistic renewal equations, Daphnia consumer-resource dynamics, and Ricker population models validate the framework.

Chapter 5: *Application of SINDy for a Tick-Borne Disease* demonstrates practical utility through application to SFTS transmission dynamics in Dalian Province, China. The framework discovers sparse distributed delay models directly from surveillance data. Sensitivity analyses assess model robustness to key parameters.

Chapter 6: *Comparative Analysis with Neural Network Approaches* brings together recent advances and explains where it fits sparse identification within the wider field of data-driven system identification. Comprehensive comparative analysis examines E-SINDy, P-SINDy, and NDDE methodologies across interpretability, delay parameter estimation, computational efficiency, and data requirements. The chapter helps you choose the right method for your specific problem and goals.

Chapter 7: *Conclusions* summarises principal contributions, discusses fundamental limitations, and outlines future research, applications to novel problem classes, and theoretical foundations for convergence and sample complexity.

I.7.1 Publications, preprints and papers in preparation

The content of this thesis is mostly based on the following works (some of them are currently under revision or in preparation).

- E. Bozzo, D. Breda, and M. Tanveer. “Sparse identification of time delay systems via pseudospectral collocation.” In: *IFAC-PapersOnLine* 58.27 (2024), pp. 108–113.

- D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Stochastic Delay Differential Equations." In: *J. Comput. Appl. Math.* (2025), p. 117247.
- D. Breda, X. Ji, G. Orosz, and M. Tanveer. "Data-driven methods for delay differential equations." In: *Delays and Structures in Dynamical Systems: Modelling, Analysis and Numerical Methods*. Ed. by D. Breda, R. Vermiglio, and J. Wu. CISM Lecture Notes. Submitted. Springer, 2025. arXiv: 2512.04894. URL: https://arxiv.org/abs/2512.04894.
- D. Conte, I. Santaniello, D. Breda, M. Tanveer, and R. D'Ambrosio. "A Matlab code for discovering governing dynamics in Stochastic Differential Equations using SINDy algorithm and Itô-Taylor based approximations." Submitted. 2025.
- D. Breda, M. Tanveer, and J. Wu. *Sparse identification of delay equations with distributed memory*. 2025. arXiv: 2512.21070 [math.DS]. URL: https://arxiv.org/abs/2512.21070.
- D. Breda, M. Tanveer, J. Wu, and X. Zhang. *Exploring Memory Effects: Sparse Identification in Vector-Borne Diseases*. 2026. arXiv: 2601.20591 [math.DS]. URL: https://arxiv.org/abs/2601.20591

I.7.2 Conference presentations

The research underlying this thesis has been presented at various international conferences and invited talks through both oral presentations and poster contributions:

Conference/Invited talks

- "Sparse Identification of Dynamical Systems with Distributed Delays." In: *IFAC Joint Conference on System Structure and Control (SSSC), Time Delay Systems (TDS), and Control of Complex Systems (COSY)*. Conference talk. France, 2025
- "Sparse Identification of Dynamical Systems with Distributed Delays." In: *DSABNS 2025, 16th Dynamical Systems applied on Biology and Natural Sciences*. Conference talk. Italy, 2025
- "Sparse Identification of Time Delay Systems via Pseudospectral Collocation." In: *TDS 2024, 18th IFAC Workshop on Time Delay System*. Conference talk. Italy, 2024
- "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *SciCADE 2024, International Conference on Scientific Computation and Differential Equations*. Conference talk. Singapore, 2024
- "A Neural Network Approach to Learn Delay Differential Equations via Pseudospectral Collocation." In: *ECCOMAS CONGRESS 2024, 9th European Congress on Computational Methods in Applied Sciences and Engineering*. Conference talk. Portugal, 2024

- "A Neural Network Approach to Learn Delay Differential Equations via Pseudospectral Collocation." In: *PINN-PAD: Physics Informed Neural Networks in Padova*. Conference talk. Italy, 2024
- "Data-Driven Approaches for Structured Populations." In: Invited talk. Toronto, Canada, 2025
- "Sparse Identification of a Distributed Delay Model for a Tick-Borne Disease." In: Invited talk. Toronto, Canada, 2025

Poster presentations

- D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *IDEAS2024 - Investigating Dynamics in Engineering and Applied Science*. Hungary, 2024
- D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *Summer school NUMRAD24 - Numerical Methods for Random Differential Models*. Switzerland, 2024

I.7.3 Research funding acknowledgment

This doctoral thesis is conducted within the framework of the research project *"Computational Methods for Analysing the Dynamics of Certain Diseases"*, funded through the Italian Ministry of University and Research (MUR) through a PhD grant PNRR DM351/2022 and DM352/2022 (CUP: G23C22001320003). The goal of the project is to develop noval computational methods for investigating mathematical disease models. This thesis directly contributes to this research aim by developing data-driven methodologies for delay systems. The application to SFTS transmission dynamics (Chapter 5) demonstrates the direct utility of these methods for analysing real disease data and extracting biologically meaningful temporal structures.

1

Mathematical foundations

Contents

1.1 Introduction

This chapter collects the mathematical foundations and methodological frameworks essential to the data-driven identification techniques developed in subsequent chapters. The thesis focuses on discovering governing equations of dynamical systems from observational data, with particular emphasis on systems exhibiting memory effects through various types of delays: discrete, stochastic delay, and distributed. The preliminaries presented here provide the theoretical groundwork for applying the SINDy framework to these complex infinite-dimensional systems.

1.2 Delay differential equations

DDEs extend ODEs by incorporating memory effects, where evolution depends on both present state and history. This generates infinite-dimensional dynamical systems capable of capturing complex phenomena that ODEs cannot. Unlike finite-dimensional ODEs, identifying DDEs from data is challenging when delays are unknown, motivating the specialised techniques developed in this thesis.

For a full theoretical treatment of DDEs see, e.g., [47, 62]. Here we summarise the basic aspects of related initial value problems (IVPs) in order to introduce their description in terms of dynamical systems. In this respect, we also recall the main ingredients of the relevant semigroup theory from [50], see also Chapter 3 of [17].

DDE

For $\tau > 0$ real and n a positive integer, let $\mathcal{X} := C([-\tau, 0], \mathbb{R}^n)$ be the space of continuous functions $[-\tau, 0] \to \mathbb{R}^n$, which is a Banach space when equipped with the uniform norm $\|\varphi\|_{\mathcal{X}} := \max_{s\in[-\tau,0]} \|\varphi(s)\|$ for $\|\cdot\|$ is any norm in $\mathbb{R}^n$. A general nonlinear autonomous DDE takes the form

$$x'(t) = f(x_t), \tag{1.2.1}$$

where the RHS $f : \mathcal{X} \to \mathbb{R}^n$ governs the dynamics of the system and the *history function* $x_t \in \mathcal{X}$ at time t is defined as $x_t(s) := x(t+s)$, $s \in [-\tau, 0]$, following the standard notation [85].

Here, τ represents the maximum delay, meaning that the current evolution of the system depends on its past over the interval $[-\tau, 0]$. An IVP for (1.2.1) is specified as

$$\begin{cases} x'(t) = f(x_t), & t \geq 0, \\ x(s) = \varphi(s), & s \in [-\tau, 0], \end{cases} \tag{1.2.2}$$

for $\varphi \in \mathcal{X}$ playing the role of the initial history, viz., the state x_0 at time 0. The RHS f is assumed to be sufficiently smooth to ensure that (1.2.2) has a unique solution depending smoothly on φ. For general well-posedness results, see, e.g., [8].

It is clear from (1.2.2) that DDEs generate infinite-dimensional dynamical systems on the state space $\mathcal{X}$. Due to this infinite-dimensional nature, DDEs can exhibit complex and rich dynamics already in the scalar case [94, 128]. In the following, we refer to n as to the *physical* dimension (i.e., the number of equations) and to the infinite dimension of the state space $\mathcal{X}$ as to the *dynamical* dimension (i.e., the classical "amount" of information needed to progress with the time evolution). Next, we describe the dynamical systems framework by recalling the main tools from semigroup theory.

1.2.1 Semigroup framework

A well-posed IVP (1.2.2) allows one to introduce the solution operator $T(t) : \mathcal{X} \to \mathcal{X}$ at time $t \geq 0$ as $T(t)\varphi = x_t$, which associates to the initial state φ the corresponding history at time t. $T(t)$ acts as a translation operator, with extension rule defined through (1.2.1).

The collection $\{T(t)\}_{t\geq 0}$ forms a strongly continuous semigroup of bounded operators. Consequently, we can define the relevant infinitesimal generator $\mathcal{A} : \mathcal{D}(\mathcal{A}) \subseteq \mathcal{X} \to \mathcal{X}$ as the unbounded operator given by

$$\mathcal{A}\varphi = \varphi', \qquad \mathcal{D}(\mathcal{A}) = \{\varphi \in \mathcal{X} : \varphi' \in \mathcal{X} \text{ and } \varphi'(0) = f(\varphi)\}. \tag{1.2.3}$$

Note that the underlying DDE (1.2.1) affects only the boundary condition in $\mathcal{D}(\mathcal{A})$. Indeed, the latter represents the above mentioned rule of extension, while the differentiation action is, as known, the generator of (continuous) time translation. Consequently, (1.2.2) is equivalently recast as the *abstract* IVP

$$\begin{cases} u'(t) = \mathcal{A}u(t), & t \geq 0, \\ u(0) = \varphi, \end{cases} \tag{1.2.4}$$

in the function space $\mathcal{X}$, where the equivalence relies on $u(t) = x_t$ (to be precise, the equivalence holds when $\varphi \in \mathcal{D}(\mathcal{A})$, otherwise one can talk about *mild* solutions).

1.2.2 DDEs with discrete delays

Although the formulation (1.2.1) is general, many practical applications involve DDEs with a finite number k of constant discrete delays $0 < \tau_1 < \cdots < \tau_k$. Specifically, these equations are described as

$$x'(t) = f(x(t), x(t - \tau_1), \ldots, x(t - \tau_k)), \tag{1.2.5}$$

where, with slight abuse of notation, we now consider $f : \mathbb{R}^{n(k+1)} \to \mathbb{R}^n$. Some examples are given next.

Example 2.2: Logistic delay equation

The *logistic delay equation*, also known as the Hutchinson equation [69], is given by

$$x'(t) = rx(t)\left(1 - \frac{x(t-\tau)}{K}\right), \tag{1.2.6}$$

where $r > 0$ is the intrinsic growth rate of the population, $K > 0$ is the carrying capacity and $\tau > 0$ represents the time delay in the feedback mechanism. This model exhibits a Hopf bifurcation at $r\tau = \pi/2$ and rich dynamical behaviour, including oscillatory solutions and limit cycles.

Example 2.3: Mackey-Glass equation

The *Mackey-Glass equation* [94] is the nonlinear DDE

$$x'(t) = \beta \frac{x(t-\tau)}{1+(x(t-\tau))^{\alpha}} - \gamma x(t), \tag{1.2.7}$$

with β, γ and α positive real numbers. It was initially introduced to model the generation of blood cells in physiology, based on a feedback mechanism that relies on a delayed Hill-type response characterised by the Hill coefficient α, represented by the term

$$h(t-\tau;\alpha) := \frac{1}{1+(x(t-\tau))^{\alpha}}. \tag{1.2.8}$$

The model (1.2.7) exhibits rich dynamical behaviour that depends critically on its parameter values, including periodic solutions and chaotic dynamics. This model is particularly suitable for our investigation due to its nonlinear rational term (1.2.8), which represents the characteristic feedback mechanisms prevalent in biological systems. The rational nonlinearity represents a fundamental feature of such systems and motivates its explicit incorporation into the SINDy framework presented in this thesis. By including this model, we demonstrate the capability of our approach to recover underlying dynamics from observational data despite the scalar structure of the model. To further validate the robustness of our methodology, we consider the case where both the delay parameter τ and the feedback coefficient α are treated as unknown and must be estimated from data.

Example 1.2.2: Two-Neuron model

The *two-neuron model* with multiple time delays

$$\begin{cases} x_1'(t) = -\kappa x_1(t) + \beta \tanh(x_1(t-\tau_s)) + a_{12} \tanh(x_2(t-\tau_2)), \\ x_2'(t) = -\kappa x_2(t) + \beta \tanh(x_2(t-\tau_s)) + a_{21} \tanh(x_1(t-\tau_1)) \end{cases} \tag{1.2.9}$$

is originally introduced by [120] to describe a simple neural network consisting of two interconnected neurons. Here, $x_1(t)$ and $x_2(t)$ represent the activity of the two neurons at time t, while $\kappa > 0$ denotes the decay rate, modelling the natural dissipation of neural activity over time. The parameter β represents the self-feedback strength of each neuron through a delayed activation function and a_{12}, a_{21} are the coupling coefficients, capturing the interaction strength between the neurons. The function tanh serves as an activation function that models the nonlinear response of the neurons. Finally, τ_s represents the delay in self-feedback, while τ_1, τ_2 denotes the delays between neurons.

1.3 Stochastic differential equations

This section presents stochastic systems. We start by reviewing standard stochastic differential equations before extending the concept to include time delays.

1.3.1 Stochastic ordinary differential equations

SODE

SODEs describe the evolution in time $t \geq 0$ of a real random variable $X(t) \in \mathbb{R}^n$ through the interplay of deterministic and stochastic forces. In its classical formulation, a SODE reads

$$\mathrm{d}X(t) = f(X(t))\,\mathrm{d}t + g(X(t))\,\mathrm{d}W(t), \tag{1.3.1}$$

where $W(t)$ is a q-dimensional standard Wiener process, the *drift* function $f : \mathbb{R}^n \to \mathbb{R}^n$ governs the deterministic evolution, while the *diffusion* function $g : \mathbb{R}^n \to \mathbb{R}^{n\times q}$ controls the intensity and structure of random fluctuations.

We assume the *covariance* matrix $G := gg^T/2 \in \mathbb{R}^{n\times n}$ to be positive definite. Of course (1.3.1) is a symbolic notation for

$$X(t) = X(0) + \int_0^t f(X(\sigma))\,\mathrm{d}\sigma + \int_0^t g(X(\sigma))\,\mathrm{d}W(\sigma),$$

where the last integral is intended in the Itô sense. Under the standard global Lipschitz and linear growth conditions

$$\begin{aligned}\|f(x) - f(y)\| + \|g(x) - g(y)\| &\leq L\|x - y\|, \quad x, y \in \mathbb{R}^n,\\ \|f(x)\| + \|g(x)\| &\leq C(1 + \|x\|), \quad x \in \mathbb{R}^n,\end{aligned}$$

for L and C positive constants, the initial value problem for (1.3.1) admits a unique strong solution with continuous sample paths [102].

1.3.2 Stochastic delay differential equations

SDDE

SDDEs with a single discrete and constant delay $\tau > 0$ affecting both drift and diffusion take the form

$$\mathrm{d}X(t) = f(X(t), X(t-\tau))\,\mathrm{d}t + g(X(t), X(t-\tau))\,\mathrm{d}W(t). \tag{1.3.2}$$

In (1.3.2) drift and diffusion are described, respectively, by $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times q}$. A related initial value problem requires specifying an initial history $\varphi(t)$ on $[-\tau, 0]$ by imposing $X(t) = \varphi(t)$ for $t \in [-\tau, 0]$.

Unlike standard SODEs, solutions to SDDEs are generally non-Markovian due to their dependence on the past [95]. The initial value problem for (1.3.2) becomes well-posed under appropriate Lipschitz and linear growth conditions whose general form reads

$$\begin{aligned} \|f(\varphi) - f(\psi)\| + \|g(\varphi) - g(\psi)\| &\leq L\|\varphi - \psi\|_C, \quad \varphi, \psi \in C, \\ \|f(\varphi)\| + \|g(\varphi)\| &\leq C(1 + \|\varphi\|_C), \quad \varphi \in C, \end{aligned}$$

for drift and diffusion given as maps $C \to \mathbb{R}^n$ acting on the full history defined as a function $[-\tau, 0] \ni s \mapsto X(t)(s) := X(t+s)$ for $X(t) \in C := C([-\tau, 0]; \mathbb{R}^n)$ the classic state space of continuous functions equipped with the uniform norm $\|\varphi\|_C := \max_{s\in[-\tau,0]} \|\varphi(s)\|$ [32, 95].

1.4 Distributed delay systems

In this section, we present the mathematical framework for distributed delay systems. Specifically, we focus on two primary formulations that capture the memory effects of the system through integral functionals.

DIDE and RE

In principle, both delay integro-differential equations (DIDEs) and renewal equations (REs) capture the influence of the history of the system through integral functionals. A general DIDE is formulated as

$$x'(t) = \Gamma(x_t) \tag{1.4.1}$$

whose rate of change depends on the past, whereas a typical RE

$$x(t) = \Gamma(x_t) \tag{1.4.2}$$

directly relates the current value of the unknown to its history x_t.
In both formulations, the central component is a functional $\Gamma : X \to \mathbb{R}^n$ for n a positive integer, which processes the state history and operates in a Banach space X of functions $[-\tau, 0] \to \mathbb{R}^n$ of suitable regularity (typically continuous for DIDEs and measurable for REs [46]). Γ accounts for the aggregated influence at the current time of the history of the system on the time span $[-\tau, 0]$. This functional is in general defined by an integral

$$\Gamma(\varphi) := \int_{-\tau}^{0} \gamma(\sigma, \varphi(\sigma))\, \mathrm{d}\sigma, \quad \varphi \in X, \tag{1.4.3}$$

over the delay interval, where $\gamma : [-\tau, 0] \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth, possibly nonlinear map called the *kernel* function. The integration variable σ represents the history time from the present moment t. This *nonautonomous* dependence of γ from σ is a key aspect for its reconstruction from observed data.

Example 1.4: Logistic daphnia model

The *logistic Daphnia model* [22, 26] is given by

$$\begin{cases} b(t) = \beta S(t) \displaystyle\int_{a_*}^{a_\dagger} b(t-a)\, da, & (1.4.4a) \\ S'(t) = rS(t)\left(1 - \dfrac{S(t)}{\mathcal{K}}\right) - \nu S(t) \displaystyle\int_{a_*}^{a_\dagger} b(t-a)\, da. & (1.4.4b) \end{cases}$$

This RE coupled to a DIDE describes consumer-resource dynamics, where b denotes the birth rate of a population feeding on a resource S, with parameters β (fertility), ν (consumption rate), maturation age a_* and maximal age $a_\dagger$.

Example 1.4: Behavioural epidemiology model

In the behavioural epidemiology framework considered, e.g., in [7], the classical SIR paradigm can be expanded to include vaccination conditioned on information decision processes. Consider a stable and equally interacting population structured into compartments comprising susceptible (S), infected (I), recovered (R) and vaccinated (V) individuals, with the vaccination rate governed by an information variable $M(t)$ tracking past disease levels. A plausible mathematical model reads

$$\begin{cases} S'(t) = \mu\,[1 - p(M(t))] - \mu S(t) - \beta S(t) I(t), \\ I'(t) = \beta S(t) I(t) - (\mu + \nu) I(t), \\ R'(t) = \nu I(t) - \mu R(t), \\ V'(t) = \mu p(M(t)) - \mu V(t), \end{cases}$$

where μ is the demographic turnover rate, β is the contact rate, and ν is the recovery rate. The coverage function $p(M)$ is assumed to be non-decreasing and is used to express the proportion of immunised newborns in terms of an information variable

$$M(t) := \int_{-\infty}^{0} K(\sigma) h(S(t+\sigma), I(t+\sigma))\, \mathrm{d}\sigma.$$

Here, $h : [0, +\infty) \times [0, +\infty) \to \mathbb{R}$ accounts for the prevalence of the disease weighted through a normalised memory kernel K. Typically K has the form of an Erlang distribution, but realistic data may call for different choices asking for suitable identification tools. Note that in general the local stability at the endemic equilibrium is highly affected by the properties of K [3]: as the memory concentration increases, the system is likely to have an increased number of intervals with instability in the form of Hopf bifurcations and oscillations.

Example 1.4: Ricker-type population model

The Ricker-type population model [9, 75]

$$x'(t) = d_0 \left(\int_{-\infty}^{0} F_{n/\tau}^{(n)}(\sigma) e^{\eta + d_1\sigma - ax(t+\sigma)} x(t+\sigma)\, \mathrm{d}\sigma - x(t) \right) \tag{1.4.5}$$

models single-species population growth with a gamma-distributed maturation delay, where $x(t)$ denotes the adult population, d_0 is the death rate, and F_n represents the gamma distribution. The kernel exhibits nonautonomous dependence on both σ and state history.

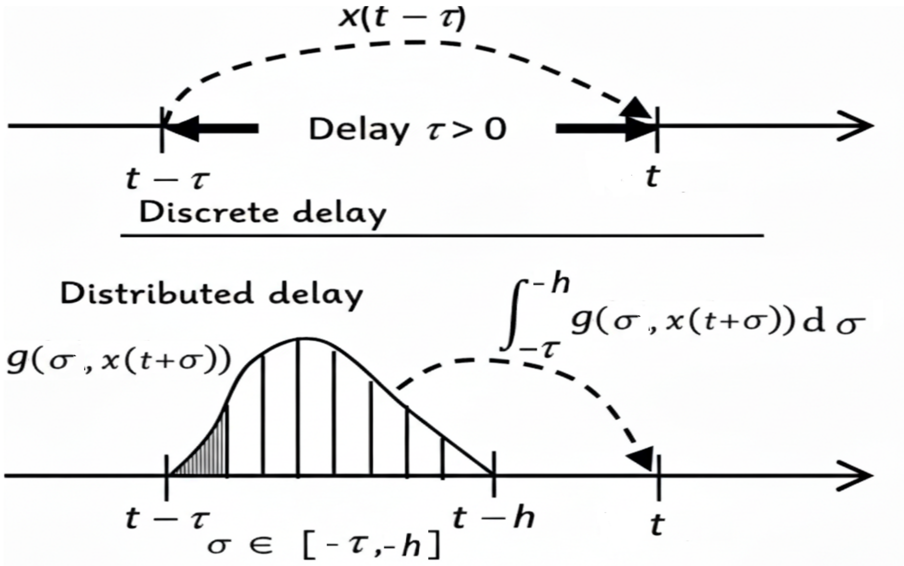

Figure 1.4.1: difference between discrete delay and distributed delay

1.5 Sparse identification of nonlinear dynamics

SINDy [30, 31, 36] is a data-driven framework to discover the governing equations of dynamical systems. Considering that most dynamical systems have an underlying sparse structure in terms of RHS description, SINDy aims to construct interpretable models from time-series data. The method assumes that the state of the system $x(t)$ is measured at discrete time points $t_1, \dots, t_m$ and its evolution is governed by the ODE

$$x' = f(x), \tag{1.5.1}$$

for an unknown RHS $f : \mathbb{R}^n \to \mathbb{R}^n$. To approximate the function f in (1.5.1), SINDy expresses each component $f_i(x)$ for $i = 1, \dots, n$ as a sparse linear combination of p candidate basis functions and $\theta_j(x)$:

$$f_i(x) = \sum_{j=1}^{p} \theta_j(x) \xi_{j,i}.$$

Here, the functions $\theta_1, \ldots, \theta_p$ serve as foundational building blocks to approximate the behaviour of the underlying system, chosen according to its expected characteristics.

1.5.1 Data matrices and libraries

The measurement data are arranged in a matrix

$$X := \begin{bmatrix} x_1(t_1) & \ldots & x_n(t_1) \\ \vdots & \ddots & \vdots \\ x_1(t_m) & \ldots & x_n(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n}, \tag{1.5.2}$$

where the rows correspond to time points and the columns correspond to state variables. Similarly, a matrix

$$X' := \begin{bmatrix} x_1'(t_1) & \ldots & x_n'(t_1) \\ \vdots & \ddots & \vdots \\ x_1'(t_m) & \ldots & x_n'(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

collects derivative values that may be directly available or estimated using numerical differentiation [37].

Although polynomial basis functions are commonly employed due to their generality and ability to capture a wide class of dynamics, other types of basis functions, such as trigonometric functions, exponentials, rational, or nonlinear functions of another type, are incorporated when the intrinsic behaviour of the system suggests their necessity.

For example, in ODEs describing oscillatory phenomena, such as the van der Pol oscillator [45] or models representing harmonic motion, trigonometric functions comprehensively describe periodic behaviours that polynomials might represent poorly or inaccurately. Similarly, exponential functions may need to properly represent the growth or decay processes that are very common in physical and biological systems.

The inclusion of specialised basis functions in the library enhances both the accuracy and interpretability of the output model, with such a choice being the most beneficial if prior knowledge or physical insight indicates a presumed importance. This flexible choice of basis functions enables SINDy to successfully characterise complicated dynamics, tailoring domain know how to select function classes that best capture the true nonlinear behaviours of the system.

We introduce the following notation to represent the system reconstruction and estimate the coefficients $\xi_{j,i}, j = 1, \ldots, p$ for each $i = 1, \ldots, n$. A library matrix $\Theta(X) \in \mathbb{R}^{m \times p}$ contains the p candidate basis functions evaluated at all m time points. For example, a polynomial library with additional trigonometric terms:

$$\Theta(X) := \begin{bmatrix} 1 & X & X^2 & \cdots & X^d & \cdots & \sin(X) & \cos(X) & \cdots \end{bmatrix}.$$

Above, X^d denotes a matrix containing all possible polynomial combinations of degree d in the components of x. For example, for $d = 2$,

$$X^2 := \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}.$$

Similarly, for trigonometric functions,

$$\sin(X) := \begin{bmatrix} \sin(x_1(t_1)) & \cdots & \sin(x_n(t_1)) \\ \vdots & \ddots & \vdots \\ \sin(x_1(t_m)) & \cdots & \sin(x_n(t_m)) \end{bmatrix}.$$

The choice of basis functions significantly affects the success of the method. As discussed in [30, 31], polynomial basis functions are commonly used to model ODEs; however, other types of functions may be more suitable, depending on the system under study. The number and type of basis functions determine the number p of columns in $\Theta(X)$. To measure the relevant complexity, either related to the model or dynamics, the total number of basis functions of a polynomial library of degree d for a system of physical dimension n is for any non-negative integers $N = n + d$ and $k = d$ where $N \geq k$, the formula is:

$$p = \binom{n+d}{d} = \frac{(n+d)!}{((n+d)-d)!d!},$$

the term $((n+d)-d)!$ simplifies simply to $n!$.

$$\binom{n+d}{d} = \frac{(n+d)!}{n!d!},$$

1.5.2 Sparse regression

To determine the coefficients $\xi_{j,i}$, we collect them into vectors $\xi_i \in \mathbb{R}^p$ for each $i = 1, \ldots, n$ and formulate the general problem as

$$X' = \Theta(X)\Xi, \tag{1.5.3}$$

where $\Xi \in \mathbb{R}^{p \times n}$ is the matrix $[\xi_1\ \xi_2\ \cdots\ \xi_n]$.

The coefficients for each column ξ_i are identified by independently solving the optimisation problem

$$\xi_i = \arg\min_{\xi \in \mathbb{R}^p} \left(\left\| X_i' - \Theta(X)\xi \right\|_2 + \lambda \left\| \xi \right\|_1 \right), \tag{1.5.4}$$

where X_i' is the ith column of X'. In the RHS of (1.5.4), the first term (in $\|\cdot\|_2$) measures the reconstruction error, whereas the second term (in $\|\cdot\|_1$) is a regularisation term that promotes sparsity through the hyperparameter $\lambda \geq 0$. Methods such as sequential thresholded least squares (STLS) [31] or least absolute shrinkage and selection operator regression (LASSO) [132] can be used to solve (1.5.4). The latter is a ℓ_1 regression analysis method that performs both variable selection and regularisation to enhance the prediction accuracy and interpretability of the model produced. The LASSO method minimises the usual sum of squared residuals with a constraint on the sum of the absolute values of the coefficients. This constraint tends to produce coefficients that are exactly zero, thereby effectively selecting the variables. Alternatively, the STLS method is used to solve optimisation problems that require sparsity. In this approach, one repeatedly solves the least squares problem and then applies a thresholding operation: any coefficient whose absolute value is below a designated threshold (given by λ) is set to zero. This iterative process is performed for a predetermined number of iterations, ultimately yielding sparse solutions. STLS is particularly effective in high-dimensional contexts where the number of features p significantly exceeds the number of observations m [31].

1.6 Pseudospectral collocation

Consider the IVP (1.2.2) and recall (1.2.3) and (1.2.4). To approximate (1.2.2) with a finite dimensional system of ODEs, we apply a *pseudospectral collocation* scheme based on *Chebyshev polynomials* [17, 133]. This method provides a systematic way to reduce the infinite dimensional state space $\mathcal{X}$ to a finite dimensional subspace $\mathcal{X}_M$ and, correspondingly, a DDE to a system of ODEs, which can then be treated with relevant ODE tools (such as the standard SINDy framework) this motivates P-SINDy as a pragmatic approach to DDE identification.

1.6.1 Chebyshev nodes and Lagrange interpolation

For a positive integer M let

$$s_i := \frac{\bar{\tau}}{2}\left(\cos\left(\frac{i\pi}{M}\right) - 1\right), \quad i = 0, 1, \ldots, M, \tag{1.6.1}$$

be the $M+1$ *Chebyshev extremal nodes* over $[-\bar{\tau}, 0]$.

Define $\mathcal{X}_M := \mathbb{R}^{n(M+1)}$ as the discrete counterpart of $\mathcal{X}$ and introduce the following *restriction* and *prolongation* operators:

$$\begin{aligned}
&R_M : \mathcal{X} \to \mathcal{X}_M, \quad R_M\psi := [\,\psi(s_0), \psi(s_1), \ldots, \psi(s_M)\,]^\intercal, \\
&P_M : \mathcal{X}_M \to \mathcal{X}, \quad P_M\Psi := \textstyle\sum_{j=0}^{M} \ell_j(s)\Psi_j.
\end{aligned}$$

The former evaluates a function $\psi \in \mathcal{X}$ at the collocation nodes, while the latter reconstructs a continuous function from its discrete values via Lagrange interpolation. The Lagrange basis functions $\{\ell_j\}_{j=0}^{M}$ are defined on the collocation nodes $\{s_j\}_{j=0}^{M}$ as

$$\ell_j(s) := \prod_{\substack{k=0 \\ k\neq j}}^{M} \frac{s - s_k}{s_j - s_k}, \quad j = 0, 1, \ldots, M, \quad s \in [-\bar{\tau}, 0]. \tag{1.6.2}$$

1.6.2 Discrete infinitesimal generator

In discrete setting, infinitesimal generator $\mathcal{A}$ is approximated by the finite-dimensional operator $\mathcal{A}_M : \mathcal{X}_M \to \mathcal{X}_M$ given by

$$[\mathcal{A}_M\Psi]_0 = f(P_M\Psi), \qquad [\mathcal{A}_M\Psi]_i = \sum_{j=0}^{M} d_{i,j}\Psi_j, \quad i = 1, \ldots, M, \tag{1.6.3}$$

where the derivatives $d_{i,j} := \ell_j'(s_i)\, I_n$, $i = 1, \ldots, M$, $j = 0, 1, \ldots, M$, of the Lagrange basis functions can be calculated explicitly and collected as entries in the *Chebyshev differentiation matrix* [133].

The discrete expressions of $\mathcal{A}_M$ above correspond to, respectively, the domain condition and the action of $\mathcal{A}$ in (1.2.3). In the resulting finite-dimensional ODE system (introduced below in (1.6.4)), the RHS f of the original DDE affects only the first equation, which is indeed a block of n ODEs. In this way, $\mathcal{A}_M$ faithfully reproduces the essential features of $\mathcal{A}$.

Note that in the resulting finite-dimensional ODE system (introduced below in (1.6.4)), the RHS f of the original DDE affects only the first equation, which is indeed a block of n ODEs. The above discretisation leads to the following finite dimensional IVP approximating (1.2.2):

$$\begin{cases} U_0'(t) = f\big(P_M\, U(t)\big), & \\ U_i'(t) = D_M\, U(t), & i = 1, \dots, M, \\ U_i(0) = \varphi(s_i), & i = 0, 1, \dots, M, \end{cases} \tag{1.6.4}$$

where

$$D_M := \begin{bmatrix} \ell_0'(s_1) & \dots & \ell_M'(s_1) \\ \vdots & \ddots & \vdots \\ \ell_0'(s_M) & \dots & \ell_M'(s_M) \end{bmatrix},$$

$$U(t) := [\, U_0(t), U_1(t), \dots, U_M(t)\,]^\intercal \in \mathcal{X}_M,$$

and each component $U_i(t)$ approximates $x_t(s_i) = x\big(t + s_i\big)$. As anticipated, $U_0(t) \approx x(t)$ is the only quantity of interest and the corresponding ODE is the only one reconstructed by SINDy. However, it is coupled to all the other components $U_i(t), i = 1, \dots, M$ through the interpolation polynomial $P_M U(t)$.

In summary, the pseudospectral collocation approach, using the Chebyshev nodes and the associated differentiation matrix, typically achieves spectral accuracy [10, 133]. For delay problems, it provides a systematic way to approximate infinite dimensional dynamics within a finite dimensional ODE framework. Note that alternative discretisation approaches can also be implemented, such as those described in [14, 87].

1.7 Itô-taylor expansions and moment estimation

The practical estimation of drift f and diffusion g functions in SODEs is fundamentally based on Itô-Taylor expansions, which provide a systematic framework for expressing increments of test functions of stochastic processes in terms of differential operators and stochastic integrals [82, 139].

1.7.1 Differential operators

For SODE (3.2.1), we define the fundamental differential operators that govern stochastic evolution. Let

$$L^0 := \sum_{i=1}^{n} f_i(x) \frac{\partial}{\partial x_i} + \sum_{i,j=1}^{n} g_{i,j}(x) \frac{\partial^2}{\partial x_i \partial x_j}, \tag{1.7.1}$$

$$L^j := \sum_{i=1}^{n} g_{i,j}(x) \frac{\partial}{\partial x_i}, \quad j = 1, \dots, q, \tag{1.7.2}$$

and

$$G_{i,j}(x) = \frac{1}{2} \sum_{k=1}^{q} g_{i,k}(x) g_{j,k}(x)$$

represents the diffusion matrix element acting on sufficiently smooth maps $\mathbb{R}^n \to \mathbb{R}$. The operator L^0 captures the combined deterministic drift and second-order stochastic corrections (the Itô correction term), while each L^j represents the first-order effects of the jth noise component.

1.7.2 Itô-Taylor expansion

For a test function ϕ the first few terms of the corresponding Itô-Taylor expansion read

$$\begin{aligned}\phi(X(t+\Delta t)) = \phi(X(t)) + L^0\phi(X(t))\Delta t + \sum_{j=1}^{n} L^j\phi(X(t))\Delta W_j(t) \\ + \frac{1}{2}(L^0)^2\phi(X(t))\Delta t + \sum_{j=1}^{n} L^j L^0 \phi(X(t)) I_{j,0} \\ + \sum_{j=1}^{n} L^0 L^j \phi(X(t)) I_{0,j} + \sum_{j,k=1}^{n} L^j L^k \phi(X(t)) I_{j,k} + \cdots\end{aligned} \tag{1.7.3}$$

where $\Delta W_j(t) := W_j(t+\Delta t) - W_j(t)$ represents the increment of the j-th component of the Brownian motion and the terms I's represent multiple stochastic integrals

$$I_{j,0} := \int_t^{t+\Delta t}\int_t^{\sigma_1} \mathrm{d}W_j(\sigma_2)\,\mathrm{d}\sigma_1,$$

$$I_{0,j} := \int_t^{t+\Delta t}\int_t^{\sigma_1} \mathrm{d}\sigma_2\,\mathrm{d}W_j(\sigma_1),$$

$$I_{j,k} := \int_t^{t+\Delta t}\int_t^{\sigma_1} \mathrm{d}W_j(\sigma_2)\,\mathrm{d}W_k(\sigma_1).$$

can be used in weak form by taking the conditional expectation of $\phi(X(t+\Delta t))$ given $X(t)$, or in strong form by working pathways with $\phi(X(t+\Delta t))$ thus retaining stochasticity [82].

1.7.3 Kramers-Moyal estimators

Using the weak form of the Itô-Taylor expansion with $\phi(x) := x_i$, the classical first-order Kramers-Moyal (KM) estimator formula for drift is

$$f_i(x) \approx \mathbb{E}\left[\frac{X_i(t+\Delta t) - X_i(t)}{\Delta t} \mid X(t) = x\right]. \tag{1.7.4}$$

For the diffusion matrix, we apply the expansion to $\phi(x) = (x_i - X_i(t))(x_j - X_j(t))$, yielding

$$G_{i,j}(x) \approx \mathbb{E}\left[\frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t} \mid X(t) = x\right]. \tag{1.7.5}$$

In the strong (pathwise) version, (3.2.7) and (3.2.8) become

$$f_i(X(t)) \approx \frac{X_i(t+\Delta t) - X_i(t)}{\Delta t}, \tag{1.7.6}$$

$$G_{i,j}(X(t)) \approx \frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t}. \tag{1.7.7}$$

1.7.4 Extension to stochastic delay differential Equations

The extension to SDDEs follows by formally replacing the state variable $X(t) \in \mathbb{R}^n$ with an augmented state variable $Z(t) = (X(t), X(t-\tau)) \in \mathbb{R}^{2n}$. For instance, the KM estimators become

$$f_i(z) \approx \mathbb{E}\left[\frac{X_i(t+\Delta t) - X_i(t)}{\Delta t} \mid X(t) = z_1, X(t-\tau) = z_2\right], \tag{1.7.8}$$

$$G_{i,j}(z) \approx \mathbb{E}\left[\frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t} \mid X(t) = z_1, X(t-\tau) = z_2\right], \tag{1.7.9}$$

where $z = (z_1, z_2) \in \mathbb{R}^n \times \mathbb{R}^n$ with z_1 representing the current time value and z_2 the delayed counterpart. Conditioning must include both current and delayed states for consistent parameter estimation due to the non-Markovian nature of SDDEs.

2

SINDy for delay differential equations

Contents

2.1 Introduction

The previous chapter established the theoretical foundations of DDEs and their dynamical properties as far as this thesis is concerned. Building on this foundation, we now address the challenge of directly identifying the governing equations of delayed dynamical systems from observational data. Reconstructing these equations from data represents a central task in modern applied mathematics and scientific computing. However, traditional system identification methods face significant obstacles when delay effects are present, but the underlying equations remain unknown [136]. Delays introduce fundamental difficulties because they introduce memory effects that increase the dimensional complexity of the model. In practical applications, both the functional form of the dynamics and the delay parameters are typically unknown. This dual uncertainty transforms system identification into a challenging inverse problem that requires data-driven approaches capable of discovering not only the functional structure but also the delay parameters of DDEs from time-series measurements. This chapter is primarily based on the work published in [16] and the framework discussed in the book chapter [25].

2.1.1 Problem statement

Existing methods for system identification face significant limitations when applied to DDEs. Most techniques rely on substantial prior knowledge of the structure of the system and the delay parameters, which is rarely available in practical applications. When the delays are unknown, the reconstruction problem becomes ill-posed because the delay parameters enter the governing equations nonlinearly, resulting in a non-convex optimisation landscape often deeply affected by multiple local minima. Furthermore, the process is computationally expensive; unlike linear coefficients, which can be determined via a single regression step, unknown delays require an iterative search where the model library must be reconstructed and solved for every candidate delay value. Conventional regression-based approaches [38] also fail to capture the inherent sparsity in the governing equations, leading to overfitted or non-interpretable models. Hence, a robust, data-driven approach is required that can identify both the functional form and delay parameters of DDEs in an interpretable and computationally efficient manner.

The specific challenges addressed in this chapter include:

- identification of DDE systems when delay values are unknown;
- sparse recovery of governing equations from limited data (few observation points);
- computational efficiency in parameter space exploration;
- balancing model complexity with interpretability.

2.1.2 Background and related work

Data-Driven methods for dynamical systems

Data-driven discovery of dynamical systems has emerged as a powerful paradigm. Traditional methods based on first principles require extensive domain knowledge, whereas modern data-driven approaches can extract governing equations directly from measurements. The SINDy framework [30, 31] represents a major advance in this area by using sparsity-promoting regression to identify parsimonious models from data.

State of the art for DDEs

Existing data-driven methods, including extensions of the SINDy framework, face particular challenges when applied to DDEs. The method proposed by [116] and [84] extends SINDy to include delayed variables and focuses on the capabilities for the delay identification, often referred to as SINDy for DDEs or denoted here as "Expert SINDy" (E-SINDy). These approaches assume that the number and value of delays are known a priori. When this information is missing, the algorithm must rely on computationally expensive brute-force (BF) techniques. These requirements restrict its applicability to systems with unknown or multiple time delays.

Other approaches for DDE identification have been developed in various contexts. Classical methods often employ optimisation over delay parameters combined with regression for functional forms, but these methods struggle with the curse of dimensionality when multiple delays are present. Recent advances incorporate Bayesian optimisation (BO) strategies to guide the search over delay parameter space, offering improved computational efficiency and scalability compared to exhaustive search techniques [104]. Building on this, our paper [16] employed particle swarm optimisation (PS), a reliable alternative to BO, and proposed the "Pragmatic SINDy" (P-SINDy) approach, which utilises pseudospectral collocation to identify models.

Gaps in existing methods

Despite advances in data-driven system identification, several critical gaps remain.

- most methods require a priori knowledge of delay values;
- computational cost scales poorly with the number of potential delays;
- trade-offs between accuracy, efficiency, and interpretability are not well characterised;
- robustness to sampling strategies and noise levels needs improvement.

These gaps motivate the development of new approaches that can handle unknown delays efficiently while maintaining the sparsity and interpretability advantages of the SINDy framework.

2.1.3 Aim and objectives

The goal of this chapter is to develop methods that eliminate the need for prior knowledge of delay values while preserving the sparsity and interpretability of the SINDy approach. This chapter introduces two complementary strategies:

(*i*) **"Expert SINDy" (E-SINDy) with external particle swarm optimisation**: This approach combines standard SINDy for DDEs with PS to discover delay parameters. PS provides superior robustness compared to BF and BO.

(*ii*) **"Pragmatic SINDy" (P-SINDy)**: This novel method reformulates the identification problem using pseudospectral collocation, converting the DDE identification problem into an optimisation over a finite-dimensional parameter space without requiring explicit knowledge of delays. P-SINDy needs only the maximum optimal delay value.

The main objectives of this chapter are as follows.

- to review the SINDy framework and its extension to systems with known delays;
- to develop E-SINDy with efficient external optimisation for unknown delays;
- to introduce the P-SINDy method based on pseudospectral collocation;
- to compare both approaches regarding accuracy, computational efficiency, and robustness;
- to validate the methods on benchmark delayed dynamical systems.

2.1.4 Chapter organisation

The remainder of this chapter is organised as follows. Section 2.2 presents E-SINDy when delays are known, extending SINDy to incorporate delayed states. Section 2.3 discusses external optimisation strategies to identify unknown delays within the E-SINDy framework. Section 2.4 introduces Pragmatic SINDy (P-SINDy), a novel method based on pseudospectral collocation that handles unknown delays more efficiently. Section 2.5 provides detailed algorithmic implementations for both approaches. Section 2.6 presents comprehensive numerical experiments on benchmark systems, comparing the performance of the methods. Finally, Section 2.7 concludes the chapter, summarising key contributions and limitations, and previews the extension to stochastic delay systems in the following chapter.

2.2 E-SINDy: Expert SINDy for known delays

Extending SINDy to DDEs [16, 84, 104, 116] introduces additional complexities due to the dependence of the system on past values. As discussed in Section 1.2, we consider the case of DDEs with a finite number of constant discrete delays, specifically those of the form given by (1.2.5). When both the number and values of the delays are known, the SINDy framework can be extended to handle such systems by incorporating delayed copies of the state variables into library functions. Explicitly, the problem can be reformulated as

$$X' = \Theta(X, X_\tau)\Xi(\tau), \tag{2.2.1}$$

where $X_\tau = [X_{\tau_1}, \ldots, X_{\tau_k}]$, with

$$X_{\tau_i} := \begin{bmatrix} x_1(t_1 - \tau_i) & \ldots & x_n(t_1 - \tau_i) \\ \vdots & \ddots & \vdots \\ x_1(t_m - \tau_i) & \ldots & x_n(t_m - \tau_i) \end{bmatrix} \in \mathbb{R}^{m \times n}$$

for $i = 1, \ldots, k$ containing delayed samples $x(t - \tau_i)$. Note that including delays increases the dimensionality of a polynomial library, with the number of terms growing from

$$p = \binom{n+d}{d} \quad \text{to} \quad q = \binom{(k+1)n + d}{d}$$

when k delays are considered with polynomials of degree d.

In previous studies [104, 116] it is assumed that delayed samples can be directly retrieved from the measurements based on the idea that each delay corresponds to an integer multiple of a uniform sampling period.

Here, we discard this assumption by acquiring delayed samples using piecewise linear interpolation of the available measurements. This allows for the consideration of both arbitrary delays and non-uniformly distributed sampling times, thus enhancing the applicability of the approach. The augmented library matrix $\Theta(X, X_\tau) \in \mathbb{R}^{m \times q}$ is then constructed by combining the basis functions of both the current data and the delayed data, e.g.,

$$\Theta(X, X_\tau) := \begin{bmatrix} 1 & X & X_\tau & X^2 & XX_\tau & X_\tau^2 & \cdots & \sin(X) & \sin(X_\tau) & \cdots \end{bmatrix}.$$

The sparse regression problem (2.2.1) depends on the known delay values $\tau_1, \ldots, \tau_k$ and is solved column-wise again as follows:

$$\xi_i(\tau) = \arg\min_{\xi \in \mathbb{R}^q} \left(\|X_i' - \Theta(X, X_\tau)\xi\|_2 + \lambda \|\xi\|_1 \right).$$

The notation $\Xi(\tau)$ in (2.2.1) highlights the dependence of the sparse solution Ξ on the given vector of delay values $\tau := [\tau_1, \ldots, \tau_k]^\intercal$. For later use, the corresponding reconstruction error is denoted as

$$\epsilon(\tau) := \|X' - \Theta(X, X_\tau)\Xi(\tau)\|_2. \tag{2.2.2}$$

As described above, this extension of SINDy to DDEs of the form (1.2.5) assumes that both the number and values of the delays are known a priori, which is uncommon in practical applications and often relies on prior modelling knowledge. The latter observation justifies the name *E-SINDy* for the proposed approach in [16], where "E" stands for "Expert"[1]. Thus, this methodology uses prior knowledge of delays to directly recover the affected DDE. Moreover, E-SINDy aims primarily at reconstructing the RHS f, which is essential for tasks such as model interpretation, rather than for long-term predictive simulations. However, it should be noted that it is not absolutely necessary to find the exact identification of the correct RHS function. Indeed, even if the reconstructed model can reproduce the observed data, it may not be representative of the underlying governing equations of the system [104].

Example 2.2: Logistic delay equation

The *logistic delay equation*, also known as the Hutchinson equation [69], is one of the most well-known DDEs. It is given by

$$x'(t) = rx(t)\left(1 - \frac{x(t-\tau)}{K}\right), \tag{2.2.3}$$

where $r > 0$ is the intrinsic growth rate of the population, $K > 0$ is the carrying capacity and $\tau > 0$ represents the time delay in the feedback mechanism corresponding to internal competition.
The qualitative behaviour of the solution of the corresponding IVP depends on r and τ: monotonic when $r\tau < 1/e$ and oscillatory when $1/e < r\tau < \pi/2$. In both scenarios, the solution converges to the unique nontrivial equilibrium point $\bar{x} = K$ as $t \to +\infty$ and K also determines how fast the system converges. At the critical parameter value $r\tau = \pi/2$ a Hopf bifurcation [86] occurs and for $r\tau > \pi/2$ the solution tends to a limit cycle. The limit cycle can be represented using the Lambert W function and as r increases, the structure of the cycle varies, resulting in sections with steep derivatives that are interspersed with plateaus [59].

[1] The term "expert" reflects an approach that directly incorporates prior knowledge of delays to handle DDEs, and in general, it tackles a DDE as it is. This terminology follows the convention in [24].

This chapter uses this model as a fundamental scalar example featuring a single discrete delay, making it an ideal starting point for demonstrating the effectiveness of sparse identification techniques for DDEs when the delay is assumed to be known. The relative simplicity of the equation, combined with its rich dynamical behaviour, provides an excellent testbed for validating the E-SINDy approach before considering more complex systems.

	Samples Type	
	Uniform	Random
Samples Size (m)	10	10
Training RMSE$_{x'}$		
STLS	$\mathbf{7.80 \times 10^{-15}}$	$\mathbf{4.19 \times 10^{-15}}$
LASSO	2.80×10^{-3}	2.66×10^{-3}
Testing RMSE$_{x'}$		
STLS	$\mathbf{4.62 \times 10^{-15}}$	$\mathbf{6.43 \times 10^{-15}}$
LASSO	2.61×10^{-3}	2.89×10^{-3}
Testing RMSE$_x$		
STLS	$\mathbf{4.93 \times 10^{-14}}$	$\mathbf{6.09 \times 10^{-14}}$
LASSO	6.35×10^{-1}	2.04×10^{-2}

Table 2.2.1: performance comparison of E-SINDy on (2.2.3) using STLS and LASSO methods with uniform and random samples, showing RMSE values for training and testing phases; see text for more details.

To illustrate the application of E-SINDy to DDEs, we reconstruct the logistic delay equation (2.2.3) for $r = 1.8$, $K = 10$ and $\tau = 1$ assumed to be known. We use the time series obtained by solving the IVP for (2.2.3) with history function $\varphi(s) = \cos(s)$ (via MATLAB's `dde23`) on the time window $[0, 30]$, using the data in $[0, 18]$ for training E-SINDy and the data in $[18, 30]$ for testing the obtained model. A polynomial library of degree $d = 2$ is used. To evaluate the reliability of the method, we consider both uniform and random (non-uniform) distributions for the $m = 10$ data points used for training. We assume that at these data points, both the state x and derivative x' are accurately measured. The results are presented in Table 2.2.1, where the following root mean squared errors are considered:

$$\text{RMSE}_{x'} := \|X' - \Theta(X, X_\tau)\Xi(\tau)\|_2, \tag{2.2.4}$$

$$\text{RMSE}_x := \|X - \mathcal{S}_{\text{DDE}}(\Theta(X, X_\tau)\Xi(\tau))\|_2. \tag{2.2.5}$$

The former is indeed $\epsilon(\tau)$ defined in (2.2.2), whereas in the latter, the notation $\mathcal{S}_{\text{DDE}}$ represents the DDE solver that progresses the model obtained by SINDy forward in time (for example, MATLAB's `dde23` is adopted in this chapter). The results in 2.2.1 show the efficacy of E-SINDy in reconstructing the RHS (both training and testing), as well as the closeness of the reconstructed trajectory to the true solution (only for the testing data). The tests involved both STLS and LASSO and show the superiority of the former.

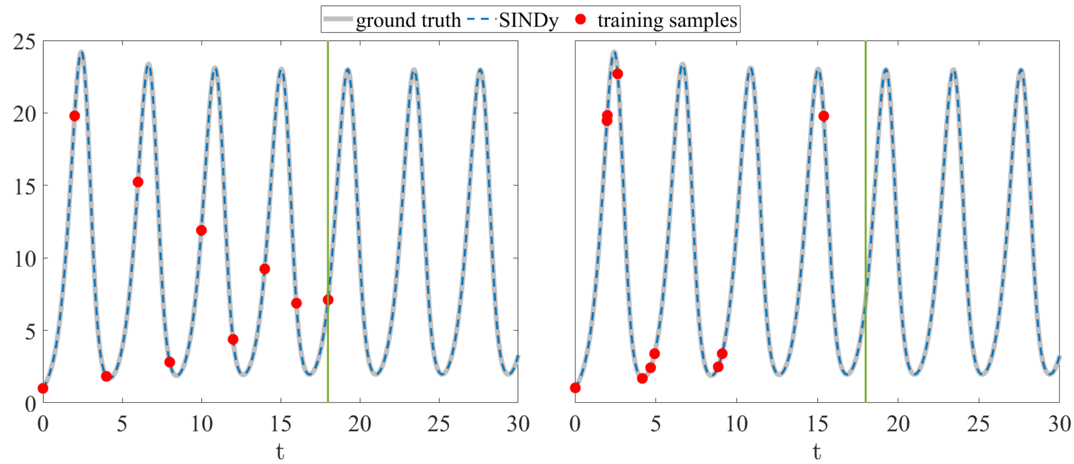

Figure 2.2.1: true (solid thick grey) and E-SINDy trajectory (dashed cyan) of the logistic delay equation (2.2.3) for $r = 1.8$, $K = 10$ and $\tau = 1$, reconstructed with $m = 10$ uniform (left) and random (right) training samples (red dots) in $[0, 18]$, and tested in $[18, 30]$.

Figure 2.2.1 presents the reconstructed trajectories for the uniform (left) and random (right) training samples, demonstrating the effectiveness of E-SINDy in identifying the governing dynamics of (2.2.3). In general, very few samples are sufficient to reconstruct the underlying model correctly.

2.3 Unknown delays and parameters

In practical scenarios, the number and values of the delay parameters in a DDE are often unknown. In the context of SINDy, one way to handle this challenge is to consider minimising the map $\tau \mapsto \epsilon(\tau)$ of the reconstruction error in (2.2.2) or (2.2.4). When only a single delay is unknown [116], a simple BF approach [51, 141] is used to evaluate $\epsilon(\tau)$ for a range of candidate delay values and select the one that minimises the reconstruction error. Although straightforward, this approach can be computationally expensive if the search space is large; the number of calls to SINDy is equal to the number of candidate delays considered.

A more sophisticated method is suggested by [104], in which BO [55, 119, 142] is used to identify the delay value that minimises $\epsilon(\tau)$, thus decreasing the number of calls to E-SINDy. In addition, [104] extended this idea to a multivariate scenario, simultaneously optimising multiple delays and/or potentially other unknown non-multiplicative parameters (e.g., non-multiplicative parameter entries in a function library). This extension is especially beneficial for cases like the Mackey-Glass equation (2.3.1), where the presence of the nonlinear rational term (2.3.2) requires identifying both τ and α.

Building on these methods, [16] presented an alternative optimisation based on a PS [13, 78, 121], which explores a continuous range for unknown delay(s) or parameter(s), offering a more efficient alternative to BO and BF, particularly in higher-dimensional parameter spaces.

In summary, we stress that the "external optimisation" methods identify unknown delays and non-multiplicative parameters, while "internal optimisation" in SINDy (e.g., STLS [31]) performs sparse regression to determine the active terms in the model based on estimated parameters.

Example 2.3: Mackey-Glass equation

The *Mackey-Glass equation* [94] is the nonlinear DDE

$$x'(t) = \beta \frac{x(t-\tau)}{1+(x(t-\tau))^{\alpha}} - \gamma x(t), \tag{2.3.1}$$

with β, γ and α positive real numbers. It was initially introduced to model the generation of blood cells in physiology, based on a feedback mechanism that relies on a delayed Hill-type response characterised by the Hill coefficient α, represented by the term

$$h(t-\tau;\alpha) := \frac{1}{1+(x(t-\tau))^{\alpha}}. \tag{2.3.2}$$

Depending on the parameter values, (2.3.1) can exhibit periodic solutions or possibly chaotic dynamics. As noted above, (2.3.1) is particularly relevant due to its nonlinear rational term (2.3.2), which is representative of characteristic feedback in biological systems. Therefore, this term is explicitly incorporated into the framework for sparse identification to accurately extract the dynamics of (2.3.1) from data despite the scalar structure.

This chapter uses this model to demonstrate simultaneous optimisation of both τ and α, assuming that their values are unknown. The presence of the nonlinear rational term (2.3.2) makes this example more challenging than (2.2.3), providing a stringent test of the identification methods' ability to handle complex nonlinear feedback with unknown parameters.

We conclude this section by testing E-SINDy with external optimisation to recover both τ and α for the Mackey–Glass equation (2.3.1). Data are obtained in the time window $[0,30]$ by simulating the corresponding IVP with the initial function $\varphi(s) = \cos(s)$ and reference parameters $\beta = 4$, $\gamma = 2$ and nominal values $\tau = 1$ and $\alpha = 9.6$ for the unknown delay and Hill exponent, respectively. We use $m = 100$ training samples in $[0,18]$ and the remaining interval $[18,30]$ to test the model. The chosen polynomial library has degree $d = 2$ and includes the rational term in (2.3.2) with a non-multiplicative unknown parameter α affecting the nonlinearity of the model.

Figure 2.3.1 (left) shows the reconstruction error (2.2.4) as a function of both τ and α using a 100×100 grid in the interval $[0.1,2]$ for τ and $[0.1,20]$ for α. Figure 2.3.1 (right) shows the corresponding trajectory obtained with the model reconstructed by E-SINDy externally optimising τ and α using BF. The values obtained are $[\widehat{\tau},\widehat{\alpha}] = [1.0020, 9.5475]$. The discrepancy in the reconstructed trajectory is due to the low accuracy of the final optimal value. The results are summarised in Table 2.6.2 of Section 2.6 for BO and PS, followed by a more detailed comparison.

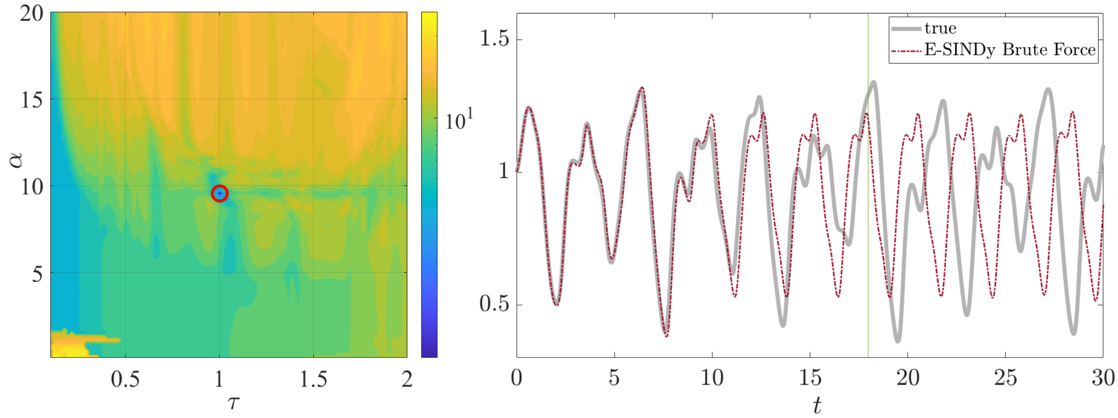

Figure 2.3.1: (left) external optimisation of the unknown delay τ and the Hill exponent α for the Mackey Glass equation (2.3.1) using BF. The colour indicates $\text{RMSE}_{x'}$ over a 100×100 grid in the (τ, α) plane, with nominal values of $(\tau, \alpha) = (1, 9.6)$, and the obtained optimal point $(\widehat{\tau}, \widehat{\alpha}) = (1.0020, 9.5475)$ is marked by a red circle. (right) True (solid thick grey) and E-SINDy (dashed red) trajectories after external optimisation.

A significant drawback of E-SINDy is represented by the assumption that the number of delays is assumed to be known, or at least the adopted library should allow for a sufficiently large number of delayed terms. As such information is often unavailable in practice, the methodology illustrated in the next section aims to address the issue by operating without requiring prior knowledge of the number of delays.

2.4 P-SINDy: Pragmatic SINDy via pseudospectral collocation

E-SINDy for unknown delays and parameters, as introduced in Section 2.3 relies on the external optimisation (through BF, BO or PS) of a predefined number $\bar{k}$ of delays, assuming $\bar{k} \geq k$, where k is the true number of delays in (1.2.5). If $\bar{k} > k$ and E-SINDy succeed, the resulting sparse matrix Ξ contains $\bar{k} - k$ null coefficients. However, multivariate optimisation is expensive in terms of computation.

To solve this issue, [16] introduced a more efficient approach where only the maximum delay in (1.2.5) needs external optimisation. This method, termed P-SINDy where P stands for "Pragmatic"[2], focuses first on reducing the problem to a finite-dimensional system of ODEs through pseudospectral collocation [22] and then applies SINDy to the resulting ODE.

Unlike E-SINDy, which relies on prior knowledge of the delay parameters to reconstruct an interpretable RHS, P-SINDy adopts a black-box approach, aiming only to match the observed trajectory and then simulate beyond the measurement time span. While P-SINDy sacrifices the interpretability of the reconstructed RHS, it requires no prior information on the number or values of delays and thus achieves enhanced computational efficiency by possibly optimising only the maximum delay $\bar{\tau} := \tau_k$ in the case where the

[2]This terminology again follows the naming convention in [24] where "Pragmatic" refers to first approximating a DDE with an ODE and then applying techniques for ODEs.

latter is unknown.

Note that in P-SINDy, the infinite-dimensional dynamical system (1.2.4) in the state space $\mathcal{X}$ is reduced to a system of ODEs, each of which is itself a system of dimension n corresponding to the original physical dimensions. Meanwhile, the actual dimension of the sparse regression problem remains confined to n, similar to E-SINDy, because only the first block (or depends on the number of state variables) of the ODEs must be reconstructed, as illustrated below. We now describe how to augment the SINDy library to obtain P-SINDy.

2.4.1 Extended collocation library

As already observed, most of the ODEs in (1.6.4) (i.e. the n-dimensional blocks for $i = 1, \ldots, M$) arise from the differentiation action of $\mathcal{A}$ and are thus independent of the specific RHS f. The maximum delay $\bar{\tau}$ only affects the length of the collocation domain $[-\bar{\tau}, 0]$. As the only ODE directly affected by f is the first block (or depends on the number of state variables)

$$U_0'(t) = f(P_M U(t)),$$

P-SINDy applies sparse regression solely to this block. The resulting regression problem reads

$$X' = \Theta(X, X_s)\Xi_M(\bar{\tau}),$$

where now $X_s = [X_{s_1}, \ \ldots \ , X_{s_M}]$ with

$$X_{s_i} := \begin{bmatrix} x_1(t_1 + s_i) & \ldots & x_n(t_1 + s_i) \\ \vdots & \ddots & \vdots \\ x_1(t_m + s_i) & \ldots & x_n(t_m + s_i) \end{bmatrix},$$

for $i = 1, \ldots, M$ and the notation $\Xi_M(\bar{\tau})$ highlights the dependence of the sparse solution on the maximum delay $\bar{\tau}$ and the degree of collocation M.

Consequently, P-SINDy operates without requiring knowledge of the possible intermediate delays. Indeed, when $k > 1$, the effect of multiple delays is mixed by interpolation, making it difficult to recover RHS f in an interpretable manner. Instead of directly identifying RHS, P-SINDy prioritises accurate trajectory reconstruction, reducing the problem of optimising only the maximum delay $\bar{\tau}$, as opposed to k delays in E-SINDy. This convenient pragmatic approach uses well-established tools for ODEs, making it a computationally efficient alternative. Note finally that the size of a corresponding polynomial library of degree d becomes

$$r = \binom{(M+1)n + d}{d}.$$

Example 2.4.1: Two-Neuron model

The *two-neuron model* with multiple time delays

$$\begin{cases} x_1'(t) = -\kappa x_1(t) + \beta \tanh(x_1(t-\tau_s)) + a_{12} \tanh(x_2(t-\tau_2)), \\ x_2'(t) = -\kappa x_2(t) + \beta \tanh(x_2(t-\tau_s)) + a_{21} \tanh(x_1(t-\tau_1)) \end{cases} \tag{2.4.1}$$

was originally introduced by [120] to describe a simple neural network consisting of two interconnected neurons. Here, $x_1(t)$ and $x_2(t)$ represent the activity of the two neurons at time t, while $\kappa > 0$ denotes the decay rate, modelling the natural dissipation of neural activity over time. The parameter β represents the self-feedback strength of each neuron through a delayed activation function and a_{12}, a_{21} are the coupling coefficients, capturing the interaction strength between the neurons. The function tanh serves as an activation function that models the nonlinear response of the neurons. Finally, τ_s represents the self-feedback delay, while τ_1, τ_2 denote the interaction delays between neurons.

We considers this model 2.4.1 as an advanced test case because it exemplifies a system of physical dimension $n = 2$ with multiple discrete delays, thus providing a robust benchmark to assess the performance of sparse identification techniques in more complex scenarios. The presence of three distinct delay parameters and nonlinear coupling through tanh makes this system considerably more challenging than the scalar examples 2.2.

We conclude this section with tests on the two-neuron system (2.4.1) of two DDEs with three delays. For reference parameters, we use $\kappa = 0.5$, $\beta = -1$, $a_{12} = 1$ and $a_{21} = 2$, with unknown delays having the nominal values $\tau_s = 1.5$, $\tau_1 = 2$ and $\tau_2 = 2$. The corresponding IVP with constant initial history $\varphi(s) \equiv [\,0.5, -0.5\,]^\intercal$ is simulated in the time window $[0, 30]$, producing $m = 100$ uniform samples in $[0, 18]$ for training the model. We use a polynomial library of degree $d = 2$ along with trigonometric functions: the polynomial terms address generic nonlinearities, whereas the trigonometric functions improve the modelling of oscillatory phenomena, as frequently seen in neural systems.

In E-SINDy with BO, the delay $\widehat{\tau}_s$ is optimised within the range $[1, 2]$ and the delays $\widehat{\tau}_1$ and $\widehat{\tau}_2$ are optimised within the range $[1, 2.5]$. In P-SINDy with only PS, only $\bar{\tau} = \widehat{\tau}_2$ is optimised using the range indicated above.

Figure 2.4.1 (left) illustrates the reconstructed trajectories for E-SINDy with BO and P-SINDy with PS and $M = 10$. Instead, Figure 2.4.1 (right) illustrates the evaluation cycle of the unknown delays obtained using PS with E-SINDy. The process is plotted against the number of iterations, which represents the number of times E-SINDy is called. Further results are presented in Table 2.6.4 in the next section. Additional results with P-SINDy are presented in Section 2.6 where the following root mean squared errors are considered:

$$\mathrm{RMSE}_{x'} := \|X' - \Theta(X, X_s)\Xi_M(\bar{\tau})\|_2, \tag{2.4.2}$$

$$\mathrm{RMSE}_{x} := \|X - \mathcal{S}_{\mathrm{ODE}}(\Theta(X, X_s)\Xi_M(\bar{\tau}))\|_2. \tag{2.4.3}$$

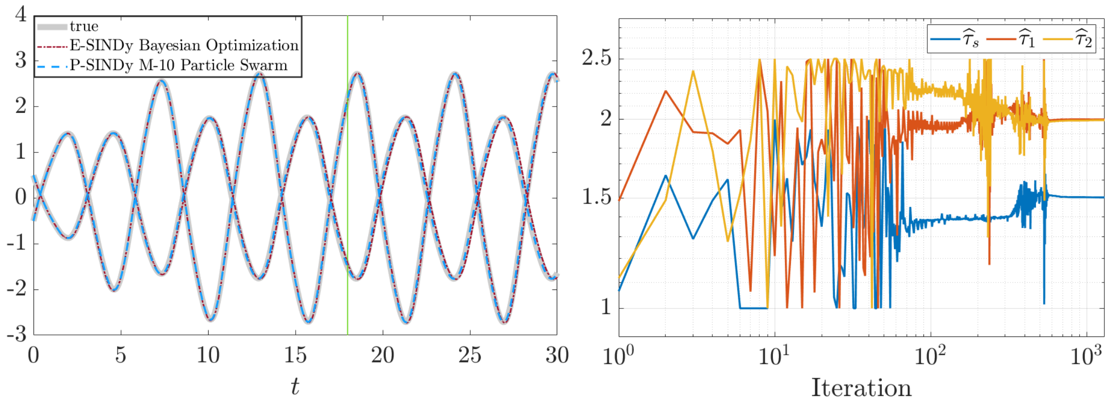

Figure 2.4.1: (left) reconstructed trajectory for the two neuron model (2.4.1) showing the true trajectory (solid thick grey), E-SINDy trajectory obtained with BO (dashed red), P-SINDy trajectory of collocation degree $M = 10$ obtained with PS (blue). The true delay values are $(\tau_s, \tau_1, \tau_2) = (1.5, 2, 2)$, the optimised delay values for E-SINDy are $(\widehat{\tau}_s, \widehat{\tau}_1, \widehat{\tau}_2) = (1.5136, 1.9478, 2.0598)$ while for P-SINDy the recovered delay is $\bar{\tau} = 2.0019$. (right) delays recovery for E-SINDy $(\widehat{\tau}_s, \widehat{\tau}_1, \widehat{\tau}_2)$ using PS. (for full comparison see Section 2.6)

Above, $\mathcal{S}_{\text{ODE}}$ computes the trajectory of the full system (1.6.4), integrating the reconstructed first block of n ODEs with the addition of the known differentiation terms from (1.6.4) (in this study, we use MATLAB's `ode45`).

2.5 Algorithmic implementation

This section provides an overview of the algorithmic implementation of the SINDy approaches for discrete delays developed in this chapter. Detailed step-by-step pseudocode and complete algorithmic descriptions are presented in Appendix A, which is dedicated to the computational implementation of all methods discussed in this thesis. The present section summarises the key algorithmic components and their computational characteristics to facilitate understanding the practical implementation of the methods.

Algorithm A.1 outlines the E-SINDy approach for cases in which all discrete delays considered relevant are known a priori. It starts with the preparation of delayed data by performing linear interpolation of the initial time series at adequately shifted temporal locations. An augmented library is then formulated by combining current and delayed variables, followed by sparse regression. The resulting sparse coefficient matrix finally defines the interpretable RHS of the governing equation.

The second option, described in Algorithm A.2 which outlines the P-SINDy approach, assumes a broader and more general context that considers only the relevant maximum delay, which is assumed unknown. Subsequently, the pseudospectral collocation grid is used over the potential delay interval using Chebyshev nodes, allowing the delayed regressors to be assembled across multiple collocation points.

As outlined in Algorithm A.3, a framework for outer loop optimisation is utilised to systematically explore parameter spaces, which may consist of either a vector of discrete delays or a single maximum delay along with the unknown non-multiplicative parameters while simultaneously monitoring the performance metrics. Various optimisation methodologies are used. The resultant outcome is a well-founded selection of model parameters, along with their corresponding sparse representations, combining computational efficiency and model accuracy.

2.6 Further tests and discussion

The scope of this section is to provide further experiments from DDEs (2.2.3), (2.3.1), and (2.4.1) to evaluate the performance of both E-SINDy and P-SINDy by employing BF, BO and PS as external optimisers for unknown delays and parameters. All the tests are run using integrated MATLAB optimisers and computational environment described in Section I.4 of the Introduction. Specifically, BO and PS optimisation are utilised from the Statistics and Machine Learning Toolbox and Global Optimisation Toolbox, respectively, with detailed implementation instructions.

Data for each experiment consisted of uniformly spaced samples m obtained by solving the corresponding DDE IVP (1.2.2) using MATLAB's `dde23` during the interval $[0, T]$ with a given history function φ. In each case, 60% of the samples are used for training and 40% for testing. In the following tables and figures, we label each SINDy approach as "E" for E-SINDy and "P" for P-SINDy, specifying the degree of collocation as P5, P10 and P15 for $M = 5$, 10 and 15, respectively. Moreover, the recovered values for the DDE coefficients, as well as for possible unknown delays and parameters, are denoted with a hat. Tables 2.6.1 to 2.6.3 report the relevant absolute errors rather than the values themselves (e.g. $|\tau - \hat{\tau}|$ rather than $\hat{\tau}$), to better highlight the accuracy of the various reconstructions.

Regarding reconstruction errors, we refer to (2.2.4) and (2.2.5) for E-SINDy and to (2.4.2) and (2.4.3) for P-SINDy. Finally, note that the use of uniformly spaced samples does not restrict the values of unknown delays, as delayed and collocation samples are reconstructed via piecewise linear interpolation if they are not available.

The first series of tests runs on example (2.2), assuming, by default, $K = 1$. The growth parameter and nominal delay values are set to $r = 1.8$ and $\tau = 1$, the latter assumed to be unknown and thus externally optimised. Furthermore, we use $m = 100$ and $T = 30$, with initial function $\varphi(s) = \cos(s)$, $s \in [-1, 0]$, and a polynomial library with degree $d = 2$. The latter describes the RHS through the vector

$$\Xi = [0\ r\ 0\ 0\ -r\ 0]^{\mathsf{T}}$$

corresponding to the library of candidate functions,

$$\Theta(X, X_\tau) = \begin{bmatrix} 1 & X & X_\tau & X^2 & XX_\tau & X_\tau^2 \end{bmatrix}. \tag{2.6.1}$$

Table 2.6.1 summarises the results for E, P5, P10 and P15 with BF, BO and PS, including the absolute errors on both r (as a measure of the goodness of the recovered sparse vector Ξ) and τ, as well as the reconstruction errors $\mathrm{RMSE}_{x'}$ and RMSE_{x}. In all cases, the external optimisers BF, BO, and PS successfully identified the correct delay τ, with slightly varying accuracy and accuracy of BF depending on the density of its candidate set (here, 1 000 values in the range $[0.1, 2]$). Moreover, BO stops after a predetermined number of calls to SINDy (300 in this case), whereas PS terminates when the desired accuracy threshold is reached (set to 10^{-3}). As both E-SINDy and P-SINDy with different optimisers recover coefficients and delays with good accuracy, the resulting trajectories for training and testing are nearly identical, and we omit plotting them, as shown in Figure 2.2.1 in Section 2.2.

The number of calls to SINDy and the corresponding CPU times are reported in the first block of rows in Table 2.6.4. Note that PS generally requires fewer calls to SINDy and consequently a much lower CPU time, yet provides the same or even better accuracy, and that the number of calls to SINDy decreases for P-SINDy with increasing M, providing, in general, better accuracy but higher CPU time. Compared to E-SINDy and BF, P5 or P10 with PS appear to be the most effective.

The second series of tests concerns (2.3) with $\beta = 4$ and $\gamma = 2$, externally optimising the unknown delay τ and the Hill exponent α of nominal values of 1 and 9.6, respectively. We use $m = 100$ and $T = 30$, with the initial function $\varphi(s) = \cos(s)$ for $s \in [-1, 0]$, along with a polynomial library of degree $d = 2$ that incorporates the Hill function in (2.3.2). The representation of the RHS is depicted through the vector

$$\Xi = [0 \ -\gamma \ 0\ 0\ 0\ 0\ 0\ 0 \ \beta \ 0]^\mathsf{T}$$

according to the library of candidate functions.

$$\Theta(X, X_\tau, H) = \begin{bmatrix} 1 & X & X_\tau & H & X^2 & XX_\tau & XH & X_\tau^2 & X_\tau H & H^2 \end{bmatrix}, \tag{2.6.2}$$

where H is the matrix of samples of the Hill function in (2.3.2). Figure 2.6.1 (left) compares the error $\mathrm{RMSE}_{x'}$ from BO and PS as a function of the total number of SINDy calls. In all three optimisers, the true values of τ and α are accurately determined, although the accuracy of BF depends on the spacing of its candidate set, while BO and PS had different stopping criteria. Figure 2.6.1 (right) illustrates the trajectory reconstructions using E-SINDy with BO and P10 with PS, respectively. The corresponding absolute errors for these final optimised values, along with the RMSE errors, are listed in Table 2.6.2.

Similarly, we collect data for the two-neuron model (2.4.1) and set the parameter values to $\kappa = 0.5$, $\beta = -1$, $a_{12} = 1$, and $a_{21} = 2$, with unknown delays of nominal values $\tau_s = 1.5$, $\tau_1 = 2$ and $\tau_2 = 2$. In addition, we choose $m = 100$ and $T = 30$ together with a constant initial function $\varphi(s) \equiv [\,0.5, -0.5\,]^\mathsf{T}$ and use a polynomial library of degrees $d = 2$ incorporating trigonometric terms. We optimised multiple delays τ_s, τ_1 and τ_2 in the two-neuron model (2.4.1). With E-SINDy, all delays are simultaneously optimised, while with P-SINDy, only the maximum delay is optimised. Figure 2.6.2 provides an overview of the optimised delays using the PS. Table 2.6.3 reports the absolute errors between the recovered and true values and the RMSE errors for BO and PS. BF is omitted because of its large computational overhead and lower accuracy in multiparameter searches.

Finally, Table 2.6.4 lists the number of SINDy calls and CPU times for each approach (E, P5, P10, and P15) combined with BF, BO, or PS. In cases involving two or more unknown parameters (as in (2.3.1) and (2.4.1)), the brute-force optimiser becomes prohibitively expensive, leading us to omit BF from some experiments. When using E-SINDy

SINDy	optimiser	$\lvert r-\widehat{r}\rvert$	$\lvert \tau-\widehat{\tau}\rvert$	$\mathrm{RMSE}_{x'}$	RMSE_x
E	BF	$\mathbf{1.00 \times 10^{-5}}$	$\mathbf{3.00 \times 10^{-4}}$	1.72×10^{-12}	6.02×10^{-5}
P5	BF	1.00×10^{-4}	$\mathbf{3.00 \times 10^{-4}}$	6.38×10^{-10}	2.43×10^{-5}
P10	BF	$\mathbf{1.00 \times 10^{-5}}$	$\mathbf{3.00 \times 10^{-4}}$	1.98×10^{-13}	$\mathbf{1.25 \times 10^{-6}}$
P15	BF	$\mathbf{1.00 \times 10^{-5}}$	$\mathbf{3.00 \times 10^{-4}}$	$\mathbf{1.10 \times 10^{-14}}$	8.83×10^{-6}
E	BO	1.00×10^{-4}	2.00×10^{-4}	3.72×10^{-11}	5.62×10^{-3}
P5	BO	1.00×10^{-4}	1.00×10^{-4}	2.68×10^{-9}	4.45×10^{-3}
P10	BO	$\mathbf{1.00 \times 10^{-5}}$	1.00×10^{-5}	9.44×10^{-11}	3.48×10^{-6}
P15	BO	$\mathbf{1.00 \times 10^{-5}}$	$\mathbf{1.00 \times 10^{-6}}$	$\mathbf{5.32 \times 10^{-12}}$	$\mathbf{6.35 \times 10^{-7}}$
E	PS	1.00×10^{-4}	$\mathbf{1.00 \times 10^{-6}}$	4.91×10^{-12}	8.17×10^{-5}
P5	PS	1.00×10^{-2}	1.00×10^{-4}	1.72×10^{-11}	2.90×10^{-5}
P10	PS	$\mathbf{1.00 \times 10^{-6}}$	1.00×10^{-5}	2.38×10^{-13}	5.14×10^{-7}
P15	PS	$\mathbf{1.00 \times 10^{-6}}$	1.00×10^{-5}	$\mathbf{4.01 \times 10^{-16}}$	$\mathbf{4.01 \times 10^{-8}}$

Table 2.6.1: performance comparison of E-SINDy (E) and P-SINDy (P, with $M = 5$, 10 and 15) for (2.2.3), showing the absolute error between the true and the recovered coefficient $|r-\widehat{r}|$ and delay $|\tau-\widehat{\tau}|$, together with $\mathrm{RMSE}_{x'}$ and RMSE_x using brute-force (BF), Bayesian optimisation (BO) and particle swarm (PS).

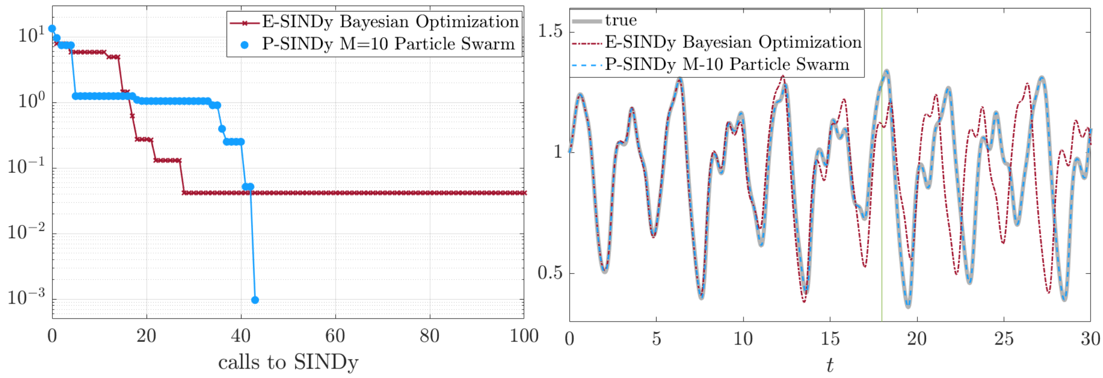

Figure 2.6.1: (left) $\mathrm{RMSE}_{x'}$ for Mackey-Glass (2.3.1) for E-SINDy and P-SINDy with $M = 10$ as a function of the total number of SINDy calls using Bayesian optimisation (BO) and Particle Swarm (PS). (right) Trajectory reconstruction for (2.3.1) showing the true trajectory (solid thick grey), E-SINDy trajectory using BO (dashed red), and P-SINDy trajectory using $M = 10$ and PS (dashed blue). For the true model, the nominal values are $(\tau, \alpha) = (1, 9.6)$. E-SINDy with BO identify $(\hat{\tau}, \hat{\alpha}) = (0.9996, 9.4969)$, whereas P-SINDy with $M = 10$ and PS identify $(\hat{\tau}, \hat{\alpha}) = (1.0000, 9.6001)$.

with multiple delays, the total cost scales with the product of the set size of candidates for each delay. In contrast, P-SINDy only optimises the maximum delay $\bar{\tau}$, considerably reducing the total computational load.

Increasing the degree of collocation M in P-SINDy increases the cost; however, we often find that a low degree, such as $M = 5$, yields satisfactory results for many DDEs. The one exception is the Mackey–Glass equation (2.3), whose potential chaotic nature demands $M = 10$ to achieve accurate trajectory reconstruction. In this case, P10-PS converged in fewer SINDy evaluations despite a similar CPU time.

In summary, both E-SINDy and P-SINDy with different external optimisers correctly recovered dynamics, identified unknown parameters and returned accurate values for both $\mathrm{RMSE}_{x'}$ and RMSE_x with the corresponding effective trajectory reconstruction for both training and testing. E-SINDy is more direct when the user already knows the structure and number of delays, while P-SINDy is more practical and, in general, computationally cheaper when multiple or unknown delays must be discovered from the data. As far as external optimisers are concerned, PS is especially effective in minimising external search costs, often outperforming both BF and BO. As a general indication, P5 and P10 with PS represent the preferred choices on average.

2.7 Conclusions

In summary, this chapter has extended/developed and validated two complementary approaches for identifying DDEs from time-series data using the SINDy framework. These methods address a fundamental challenge in data-driven system identification: discovering both the functional form and delay parameters of DDEs without requiring prior knowledge of the system structure.

SINDy	optimiser	$\lvert\xi_2-\widehat{\xi}_2\rvert$	$\lvert\xi_9-\widehat{\xi}_9\rvert$	$\lvert\tau-\widehat{\tau}\rvert$	$\lvert\alpha-\widehat{\alpha}\rvert$	$\text{RMSE}_{x'}$	RMSE_x
E	BO	$\mathbf{1.28\times^{-7}}$	$\mathbf{1.46\times10^{-7}}$	4.00×10^{-4}	1.03×10^{-1}	7.74×10^{-5}	$\mathbf{1.49\times10^{-3}}$
P5	BO	6.80×10^{-5}	4.12×10^{-4}	5.02×10^{-2}	3.67×10^{-1}	1.63×10^{-4}	9.85×10^{-2}
P10	BO	1.00×10^{-6}	1.30×10^{-5}	$\mathbf{1.00\times10^{-4}}$	1.34×10^{-2}	1.21×10^{-5}	1.64×10^{-2}
P15	BO	7.19×10^{-6}	1.00×10^{-6}	$\mathbf{1.00\times10^{-4}}$	$\mathbf{1.00\times10^{-4}}$	$\mathbf{5.45\times10^{-6}}$	6.13×10^{-3}
E	PS	1.00×10^{-6}	2.00×10^{-5}	1.00×10^{-6}	1.00×10^{-4}	6.13×10^{-7}	$\mathbf{5.45\times10^{-6}}$
P5	PS	2.06×10^{-5}	1.25×10^{-6}	6.00×10^{-4}	1.57×10^{-1}	1.64×10^{-4}	1.21×10^{-2}
P10	PS	1.03×10^{-6}	1.00×10^{-6}	2.30×10^{-6}	1.00×10^{-4}	1.09×10^{-7}	1.63×10^{-4}
P15	PS	$\mathbf{1.00\times10^{-7}}$	$\mathbf{3.00\times10^{-7}}$	$\mathbf{1.80\times10^{-7}}$	$\mathbf{1.10\times10^{-5}}$	$\mathbf{1.03\times10^{-8}}$	7.74×10^{-5}

Table 2.6.2: performance comparison of E-SINDy (E) and P-SINDy (P, with $M = 5$, 10 and 15) for (2.3.1), highlighting the absolute error between the true and recovered coefficients $|\xi_i - \widehat{\xi}_i|$, delay $|\tau - \widehat{\tau}|$ and parameter $|\alpha - \widehat{\alpha}|$, together with $\text{RMSE}_{x'}$ and RMSE_x using Bayesian optimisation (BO) and particle swarm (PS).

SINDy	optimiser	$\lvert\tau_s-\widehat{\tau}_s\rvert$	$\lvert\tau_1-\widehat{\tau}_1\rvert$	$\lvert\tau_2-\widehat{\tau}_2\rvert$	$\text{RMSE}_{x'}$	RMSE_x
E	BO	1.36×10^{-2}	5.22×10^{-2}	$\mathbf{5.92\times10^{-2}}$	7.74×10^{-5}	1.03×10^{-1}
P5	BO	-	-	1.52×10^{-1}	1.63×10^{-4}	1.09×10^{-1}
P10	BO	-	-	1.06×10^{-1}	1.21×10^{-5}	2.47×10^{-2}
P15	BO	-	-	8.27×10^{-2}	$\mathbf{5.45\times10^{-6}}$	$\mathbf{1.88\times10^{-2}}$
E	PS	2.70×10^{-3}	9.00×10^{-4}	4.90×10^{-3}	6.13×10^{-4}	2.76×10^{-2}
P5	PS	-	-	3.60×10^{-3}	1.64×10^{-6}	6.21×10^{-2}
P10	PS	-	-	1.90×10^{-3}	1.09×10^{-7}	4.85×10^{-3}
P15	PS	-	-	$\mathbf{1.00\times10^{-6}}$	$\mathbf{1.03\times10^{-7}}$	$\mathbf{1.94\times10^{-3}}$

Table 2.6.3: performance comparison of E-SINDy (E) and P-SINDy (P, with $M = 5$, 10 and 15) for (2.4.1), highlightning the absolute error between the true and optimised delays $|\tau_s - \widehat{\tau}_s|$, $|\tau_1 - \widehat{\tau}_1|$ and $|\tau_2 - \widehat{\tau}_2|$, together with $\text{RMSE}_{x'}$ and RMSE_x using Bayesian optimisation (BO) and particle swarm (PS).

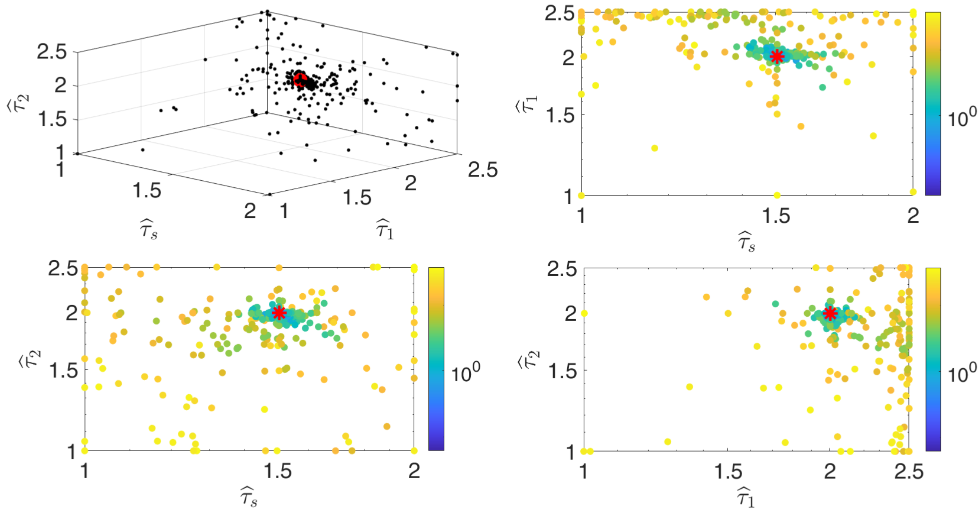

Figure 2.6.2: scatter plots of optimised unknown delays ($\widehat{\tau}_s$, $\widehat{\tau}_1$, $\widehat{\tau}_2$) using particle swarm (PS) for the two neuron model (2.4.1), colour-coded by $\text{RMSE}_{x'}$ values. Red stars mark the true values of the parameters.

DDE	SINDy	BF		BO		PS	
		Calls	Time (s)	Calls	Time (s)	Calls	Time (s)
(2.2.3)	E	1 000	204	300	494	308	**67**
	P5	1 000	217	300	438	241	70
	P10	1 000	449	300	485	179	100
	P15	1 000	1 032	300	692	**152**	163
(2.3.1)	E	10 000	hrs	300	561	284	**92**
	P5	10 000	hrs	300	682	153	94
	P10	10 000	hrs	300	1 080	**51**	102
	P15	10 000	hrs	300	1 728	245	837
(2.4.1)	E	-	-	300	723	1 303	906
	P5	-	-	300	1 081	344	**378**
	P10	-	-	300	1 410	342	494
	P15	-	-	300	2 088	**130**	580

Table 2.6.4: quantitative evaluation of computational performance of E-SINDy (E) and P-SINDy (P, with $M = 5$, 10 and 15) showing number of SINDy calls and CPU times for the three different models (2.2.3), (2.3.1), (2.4.1) and for the optimisers brute-force (BF), Bayesian optimisation (BO) and particle swarm (PS).

The main contributions of this chapter are:

- E-SINDy which assumes known delays and incorporates delayed states into the model library for sparse regression and also integrates PS optimisation to robustly estimate unknown delays and non-multiplicative parameters simultaneously.
- P-SINDy which only estimates the maximum delay using pseudospectral collocation to reduce the system to ODEs, allowing efficient sparse identification without requiring full delay knowledge.

E-SINDy accurately reconstructs the system dynamics when delays are known, using optimisation methods to find unknown delay values. However, it is computationally expensive and less practical when delays are unknown or multiple. P-SINDy overcomes this by optimising only the maximum delay and transforming the problem into a finite-dimensional ODE system, which is simpler to handle and computationally efficient. This approach sacrifices interpretability but achieves reliable predictions and trajectory reconstruction.

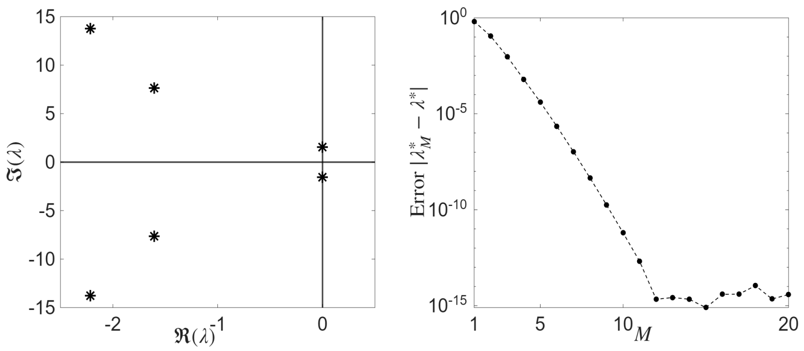

Figure 2.7.1: Preservation of dynamical properties by the P-SINDy model for $x'(t) = ax(t) + bx(t-\tau)$ at the Hopf bifurcation ($a = 0, b = -\pi/2, \tau = 1$). (left) The spectrum of the P-SINDy matrix for $M = 20$ accurately captures the pure imaginary roots $\lambda = \pm i\pi/2$. (right) The absolute error in the dominant eigenvalue decays exponentially as M increases, demonstrating spectral accuracy.

To verify that the P-SINDy ODE model preserves the original infinite-dimensional dynamical properties, we analyse the linear model:

$$x'(t) = ax(t) + bx(t-\tau)$$

setting $a = 0, b = -\pi/2$, and $\tau = 1$ the system analytically undergoes a Hopf bifurcation, with dominant eigenvalues at $\lambda = \pm i\pi/2$. Because P-SINDy relies on pseudospectral collocation, it theoretically guarantees the preservation of both the linear spectrum [27] and the local bifurcation structures [23].

We verify this by computing the eigenvalues of the obtained P-SINDy model. Figure 2.7.1 (left) shows that for $M = 20$, the P-SINDy model precisely captures the pure imaginary roots, successfully preserving the system's linear stability. Furthermore, Figure 2.7.1 (right) demonstrates that the error in this dominant eigenvalue decays exponentially as the collocation degree M increases. This confirms that P-SINDy preserves the essential stability and bifurcation properties of the original delay system with spectral accuracy.

2.7.1 Limitations and future directions

Although the methods presented in this chapter are effective for deterministic delay systems with moderate complexity, several limitations remain:

(*i*) current implementation handles single delays or a small number of discrete delays. Extension to distributed delays or continuous delay distributions requires further development;

(*ii*) automatic selection of library functions and sparsity thresholds remains challenging;

(*iii*) while P-SINDy achieves excellent trajectory matching through pseudospectral collocation, the discovered models may lack direct interpretability compared to E-SINDy;

(*iv*) deterministic framework does not account for stochastic fluctuations present in many real systems.

These limitations point to opportunities for further research, some of which are addressed in the following chapters of this thesis. In particular, the last limitation motivates the extension to stochastic delay systems developed in the next chapter.

Having established methods for deterministic delay systems, the next chapter extends the framework to SDDEs. This extension is crucial because many real-world systems exhibit both delay effects and stochastic fluctuations. The deterministic SINDy framework developed here provides the foundation for identifying drift terms in SDDEs, while new techniques are required to identify diffusion terms from noisy data.

Together, these chapters provide a comprehensive toolkit for data-driven identification of delayed dynamical systems spanning both deterministic and stochastic regimes.

Remark (Data requirements and library design): The required data volume depends strongly on the system class, noise level, and library complexity. Low-dimensional deterministic DDEs with smooth dynamics (such as the logistic delay equation in Section 2.2) can be identified from few noise-free samples, whereas the stochastic systems in Chapter 3 require many trajectories to estimate moments reliably and separate drift from diffusion. In general, sample complexity increases with the effective dimension (current and delayed states), the size and expressiveness of the candidate library, and the noise variance; when the true terms are present in the library SINDy can recover the correct model with relatively modest data, but if key terms are missing it cannot reconstruct the governing equations, making the joint study of sample complexity and library design an important direction for future research.

3

SINDy for stochastic delay differential equations

Contents

3.1 Introduction

The deterministic methods developed in Chapter 2 have proven effective for identifying equations with (discrete) delays. However, many real-world systems exhibit simultaneous stochasticity and delay effects, a combination that challenges standard identification approaches. This chapter tackles this coupled difficulty by extending SINDy to SDDEs, enabling joint identification of drift and diffusion terms from noisy observations.

SDDEs are widely used to describe systems where both stochasticity and time delays influence the evolution of state variables [4, 32, 66, 82, 101, 102]. The coexistence of random perturbations and delay effects introduces non-Markovian behaviour, making the identification of governing equations from data a challenging task. Developing reliable, interpretable, and efficient data-driven techniques for such systems is therefore an important research direction in modern applied mathematics. This chapter is primarily based on the work published in [18], from which all figures have been reproduced.

3.1.1 Problem statement

The identification of SDDEs from data requires estimating both drift and diffusion functions that depend on current and delayed states. This task is particularly difficult when these functions are unknown and must be reconstructed directly from noisy time-series data. Existing approaches [12, 64, 139] often rely on simplifying assumptions or prior knowledge of the system. They also face difficulties associated with limited data, high stochastic variance, and the inherent infinite-dimensional nature of SDDEs. Thus, there remains a need for a flexible and fully data-driven method that can recover the underlying equations without assuming pre-specified drift and diffusion structures.

The specific challenges addressed in this chapter include:

- simultaneous identification of drift and diffusion terms in the presence of delays;
- sparse recovery when both drift and diffusion admit sparse representations;
- computational efficiency for systems recovery;
- robustness to sampling strategies and stochastic noise levels;
- creating/having sufficient data.

3.1.2 Background and related work

Data-Driven methods for stochastic systems

Stochastic system identification has a rich history in the statistical and machine learning communities [76, 105]. Traditional approaches based on maximum likelihood estimation and Bayesian inference provide rigorous statistical frameworks but often require strong parametric assumptions. Modern data-driven methods seek to discover governing equations directly from data with minimal prior assumptions.

The SINDy framework has been successfully extended to SODEs [12, 42, 139]. These extensions typically employ realisation based approach to separate drift and diffusion terms.

State of the art for stochastic delay systems

The most relevant contribution is the study by [64], which extends the SINDy framework to SDDEs using a realisation-based approach. This method employs the Kramers-Moyal (KM) formalism. However, this strategy assumes prior knowledge of the drift and diffusion functional forms to generate synthetic trajectories. Consequently, it is not entirely suitable for model discovery when the governing equations are completely unknown.

In this chapter, we address this limitation by proposing a general data-driven framework for recovering drift and diffusion dynamics without such priors. This methodology and the accompanying numerical results are based on our works [18, 42].

Gaps in existing methods

Despite progress in stochastic system identification, several critical gaps remain for SDDEs:

- most methods require a priori specification of drift and diffusion functional forms;
- trade-offs between statistical efficiency and computational tractability are not well understood;
- robustness to limited data and high noise levels needs improvement.

These gaps motivate the development of new approaches that can discover sparse SDDE models from noisy data without requiring prior knowledge of functional forms.

3.1.3 Aim and objectives

The goal of this chapter is to extend the SINDy framework developed in the previous chapter to stochastic delay systems. This chapter introduces a comprehensive methodology that combines trajectory-based estimation with sparse regression to identify both drift and diffusion terms. To achieve this, the proposed framework relies on:

(*i*) using multiple noisy realisations to estimate drift (mean behaviour) and diffusion (variance behaviour) separately;

(*ii*) employing first and second moment estimators derived from the Kramers-Moyal expansion, as well as higher-order difference schemes;

(*iii*) applying sparsity optimisation to discover parsimonious representations of both terms.

Three estimation strategies are proposed based on data availability:

(*i*) **Approach A:** employs conditional sampling to generate multiple synthetic trajectories from a single observed path;

(*ii*) **Approach B1:** combines data from multiple trajectories by averaging increment statistics before sparse regression, leading to robust model discovery;

(*iii*) **Approach B2:** applies sparse regression separately to each trajectory and averages the identified models afterwards.

All strategies are compatible with higher-order estimators and can be adapted to weak or strong formulations, depending on the available data type. This modular structure allows direct estimation of drift and diffusion terms even in complex stochastic systems with delays.

The main objectives of this work are

- to develop a general data-driven framework for identifying SDDEs without requiring prior knowledge of drift and diffusion functions;
- to integrate multiple-trajectory approaches and higher-order estimation schemes into the SINDy framework;
- to assess the performance of different estimation strategies (Approaches A, B1, and B2) in terms of accuracy, robustness, and computational efficiency;
- to validate the proposed methodology through numerical experiments on SDDE models.

3.1.4 Chapter organisation

The Chapter is organised as follows. In Section 3.2, we introduce the notation and main concepts related to SODEs and SDDEs, focusing on how to recover the KM formulas and their higher-order extensions to estimate the drift and diffusion terms, based on Itô–Taylor expansions. Section 3.3 provides a concise overview of the original SINDy framework for SODEs. The discussion then moves to SDDEs in Section 3.4, highlighting how the method can be adapted to account for both the delayed and stochastic components. In Section 3.5, we discuss the practical use of the estimators introduced in Section 3.2 depending on the available data. Section 3.6 provides an overview of the algorithmic pseudocodes for all key methods, effectively connecting theoretical advancements with their practical applications. Their comparative performance is evaluated in Section 3.7, which presents a computational analysis of representative SDDE models. Finally, Section 3.8 concludes the chapter, summarising contributions, discussing limitations, and previewing the extension to distributed delay systems in the following chapter.

3.2 Estimates of drift and diffusion via Itô-Taylor Expansions

The practical estimation of drift and diffusion functions in SODEs is fundamentally based on Itô-Taylor expansions, which provide a systematic framework for expressing increments of test functions of stochastic processes in terms of differential operators and several stochastic integrals. These expansions offer a stochastic analogue of the Taylor series for deterministic ODEs, enabling the approximation of SODE solutions over short time intervals with controlled accuracy. A recent comprehensive analysis by [139] presents several higher-order extensions of the classical first-order KM formulas, originally proposed by [12] as the foundation for extending SINDy to SODEs.

SODE

SODEs describe the evolution in time $t \geq 0$ of a real random variable $X(t) \in \mathbb{R}^n$ through the interplay of deterministic and stochastic forces. In its classical formulation without delays a SODE reads

$$\mathrm{d}X(t) = f(X(t))\,\mathrm{d}t + g(X(t))\,\mathrm{d}W(t), \tag{3.2.1}$$

where $W(t)$ is a q-dimensional standard Wiener process, the *drift* function $f : \mathbb{R}^n \to \mathbb{R}^n$ governs the deterministic evolution and represents the expected direction of motion at each point, while the *diffusion* function $g : \mathbb{R}^n \to \mathbb{R}^{n\times q}$ controls the intensity and structure of random fluctuations.

We assume the *covariance* matrix

$$G := gg^T/2 \in \mathbb{R}^{n\times n}$$

to be positive definite, thus ensuring that g is non-degenerate and can be uniquely determined from G up to a right orthogonal transformation [103, 109]. The latter assumption is not restrictive, and indeed many physical systems affected by noise fit into this framework [54, 102]. Of course (3.2.1) is just a symbolic notation for

$$X(t) = X(0) + \int_0^t f(X(\sigma))\,\mathrm{d}\sigma + \int_0^t g(X(\sigma))\,\mathrm{d}W(\sigma),$$

where the last integral is intended in the Itô sense. Under the standard global Lipschitz and linear growth conditions,

$$\begin{aligned} \|f(x) - f(y)\| + \|g(x) - g(y)\| &\leq L\|x - y\|, \quad x, y \in \mathbb{R}^n, \\ \|f(x)\| + \|g(x)\| &\leq C(1 + \|x\|), \quad x \in \mathbb{R}^n, \end{aligned}$$

for L and C positive constants, the initial value problem for (3.2.1) admits a unique strong solution with continuous sample paths (for a proof, see, e.g., [102]). We observe that alternative hypotheses may also be assumed, e.g., one-sided Lipschitz continuous drift and locally Lipschitz continuous diffusion, in order to ensure bounded moments for the solution [67].

Many real-world systems exhibit memory effects, for which the current evolution of the state depends also on past values [43, 135]. In the presence of randomness, SDDEs capture this phenomenon by incorporating explicit time delays into the dynamics [32, 60, 95, 131].

SDDE

Here we consider SDDEs with a single discrete and constant delay $\tau > 0$ affecting both drift and diffusion:

$$\mathrm{d}X(t) = f(X(t), X(t-\tau))\,\mathrm{d}t + g(X(t), X(t-\tau))\,\mathrm{d}W(t). \tag{3.2.2}$$

In (3.2.2) drift and diffusion are described, respectively, by $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times q}$. A related initial value problem requires specifying an initial history through a random variable $\varphi(t)$ defined on $[-\tau, 0]$ by imposing $X(t) = \varphi(t)$ for $t \in [-\tau, 0]$.

Unlike standard SODEs, solutions to SDDEs are generally non-Markovian due to their dependence on the past [95]. Nevertheless, the initial value problem for (3.2.2) becomes well-posed under appropriate Lipschitz and linear growth conditions whose general form reads

$$\|f(\varphi) - f(\psi)\| + \|g(\varphi) - g(\psi)\| \leq L\|\varphi - \psi\|_C, \quad \varphi, \psi \in C,$$
$$\|f(\varphi)\| + \|g(\varphi)\| \leq C(1 + \|\varphi\|_C), \quad \varphi \in C,$$

for drift and diffusion given as maps $C \to \mathbb{R}^n$ acting on the full history defined as a function $[-\tau, 0] \ni s \mapsto X(t)(s) := X(t+s)$ for $X(t) \in C := C([-\tau, 0]; \mathbb{R}^n)$ the classic state space of continuous functions equipped with the uniform norm [32, 95]

$$\|\varphi\|_C := \max_{s\in[-\tau,0]} \|\varphi(s)\|.$$

As anticipated, our goal is to estimate the assumed to be unknown drift f and diffusion g from trajectory data given in terms of the m time samples $X(t_1), \ldots, X(t_m)$, possibly related to multiple paths. Note once more that one can recover g uniquely once G has been identified under the assumption that the latter is positive definite.

In this section, we present a detailed exposition of the estimation methods from [139] the main ingredients, beginning with the fundamental theoretical framework and progressing through increasingly sophisticated higher-order approximations. Mathematical development provides both the theoretical foundations and practical implementation details necessary for robust parameter estimation in stochastic dynamical systems. Later, it is extended to SDDEs and then compared in Section 3.7.

The extension to SDDEs is illustrated only at the end, relying on an obvious argument. For a complete reference on stochastic Taylor expansions for SODEs, see [82, Chapter 5]; for SDDEs, see [35, 113, 114]. We assume throughout that f and g in (3.2.1) (and then in (3.2.2)) are sufficiently smooth with bounded derivatives to ensure the validity of the considered expansions and meaningful estimates of the related remainders.

3.2.1 Mathematical framework and Itô-Taylor expansions

For the SODE (3.2.1), we define the fundamental differential operators that govern stochastic evolution. Let

$$L^0 := \sum_{i=1}^{n} f_i(x)\frac{\partial}{\partial x_i} + \sum_{i,j=1}^{n} g_{i,j}(x)\frac{\partial^2}{\partial x_i \partial x_j}, \tag{3.2.3}$$

$$L^j := \sum_{i=1}^{n} g_{i,j}(x)\frac{\partial}{\partial x_i}, \quad j = 1, \ldots, q, \tag{3.2.4}$$

be differential operators acting on sufficiently smooth maps $\mathbb{R}^n \to \mathbb{R}$, where subscripts denote components of vectors and matrices and where

$$G_{i,j}(x) = \frac{1}{2}\sum_{k=1}^{q} g_{i,k}(x) g_{j,k}(x)$$

represents the diffusion matrix element. The operator L^0 is for the combined deterministic drift motion and second-order stochastic corrections (the Itô correction term), while each L^j represents the first-order effects of the jth noise component.

For a specific test function ϕ the first few terms of the corresponding Itô-Taylor expansion read

$$\begin{aligned}\phi(X(t+\Delta t)) = {} & \phi(X(t)) + L^0\phi(X(t))\Delta t + \sum_{j=1}^{n} L^j\phi(X(t))\Delta W_j(t) \\ & + \frac{1}{2}(L^0)^2\phi(X(t))\Delta t + \sum_{j=1}^{n} L^j L^0 \phi(X(t)) I_{j,0} \\ & + \sum_{j=1}^{n} L^0 L^j \phi(X(t)) I_{0,j} + \sum_{j,k=1}^{n} L^j L^k \phi(X(t)) I_{j,k} + \cdots\end{aligned} \tag{3.2.5}$$

where $\Delta W_j(t) := W_j(t + \Delta t) - W_j(t)$ represents the increment of the j-th component of the underlying Brownian motion and the terms I's represent the multiple stochastic integrals

$$I_{j,0} := \int_t^{t+\Delta t} \int_t^{\sigma_1} \mathrm{d}W_j(\sigma_2)\,\mathrm{d}\sigma_1,$$

$$I_{0,j} := \int_t^{t+\Delta t} \int_t^{\sigma_1} \mathrm{d}\sigma_2\,\mathrm{d}W_j(\sigma_1),$$

$$I_{j,k} := \int_t^{t+\Delta t} \int_t^{\sigma_1} \mathrm{d}W_j(\sigma_2)\,\mathrm{d}W_k(\sigma_1).$$

(3.2.5) can be used in weak form by taking the conditional expectation of $\phi(X(t + \Delta t))$ given $X(t)$, or in strong form by working pathways with $\phi(X(t + \Delta t))$ thus retaining stochasticity [82].

3.2.2 Weak and strong formulations

The Itô-Taylor expansion (3.2.5) admits two fundamental interpretations that lead to different estimation strategies.

For instance, using the weak form with $\phi(x) := x_i$ for some $i = 1, \dots, n$ leads to

$$\mathbb{E}\left[x_i(t+\Delta t) \mid X(t) = x\right] = x_i + f_i(x)\Delta t + O(\Delta t^2). \tag{3.2.6}$$

This formulation eliminates the stochastic integrals involving Brownian motion increments because they have zero conditional expectation. Weak expansion provides the foundation for bias analysis of estimation methods. Using the weak expansion on the coordinate function $\phi(x) = x_i$ results in the classical first order KM estimator formula,

$$f_i(x) \approx \mathbb{E}\left[\frac{X_i(t+\Delta t) - X_i(t)}{\Delta t} \mid X(t) = x\right] \tag{3.2.7}$$

follows as a component-wise estimate of drift.

For the diffusion matrix, we apply the expansion to $\phi(x) = (x_i - X_i(t))(x_j - X_j(t))$, yielding the first-order KM estimator for the diffusion matrix for the covariance

$$G_{i,j}(x) \approx \mathbb{E}\left[\frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t} \mid X(t) = x\right] \tag{3.2.8}$$

follows by choosing $\phi(x) := (x_i - X_i(t))(x_j - X_j(t))$. In the strong (pathways) version, (3.2.7) and (3.2.8) become

$$f_i(X(t)) \approx \frac{X_i(t+\Delta t) - X_i(t)}{\Delta t} \tag{3.2.9}$$

$$G_{i,j}(X(t)) \approx \frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t}. \tag{3.2.10}$$

In both the weak and strong formulations, these standard first-order estimators are referred to as KM estimators [80, 81, 112]. The weak formulation provides consistency guarantees under mild conditions, whilst the strong formulation offers direct trajectory-wise estimates that are computationally simpler to implement.

3.2.3 Higher-Order estimation methods

Recent advances in moment-based estimation methods, as reviewed by [139], demonstrate that employing higher-order finite differences improves estimation accuracy. This section pursues second-order finite difference estimators for the reconstruction of drift and diffusion coefficients from trajectory data. The development focuses on three principal schemes: forward differences, central differences, and trapezoidal rule. For each scheme, variants exhibiting weak and strong formulations are presented and analysed below, facilitating the selection of appropriate estimators for different practical scenarios.

- **Forward differences (FD).** A second-order forward finite-difference scheme for both drift and diffusion follows by considering additional terms in the Itô-Taylor expansion (3.2.5). This scheme improves upon the first-order KM estimators by reducing bias and enhancing convergence rates.

The **weak formulations** are:

$$f_i(X(t)) \approx \frac{4\mathbb{E}[X_i(t+\Delta t) - X_i(t)] - \mathbb{E}[X_i(t+2\Delta t) - X_i(t)]}{2\Delta t}, \tag{3.2.11}$$

$$\begin{aligned} G_{i,j}(X(t)) \approx \frac{1}{4\Delta t}[&4\mathbb{E}[(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))] \\ &-\mathbb{E}[(X_i(t+2\Delta t) - X_i(t))(X_j(t+2\Delta t) - X_j(t))]]. \end{aligned} \tag{3.2.12}$$

The corresponding **strong formulations** are:

$$f_i(X(t)) \approx \frac{4(X_i(t+\Delta t) - X_i(t)) - (X_i(t+2\Delta t) - X_i(t))}{2\Delta t}, \tag{3.2.13}$$

$$\begin{aligned} G_{i,j}(X(t)) \approx \frac{1}{4\Delta t}[&4(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t)) \\ &-(X_i(t+2\Delta t) - X_i(t))(X_j(t+2\Delta t) - X_j(t))]. \end{aligned} \tag{3.2.14}$$

- **Central differences (CD).** Similarly, second order can be reached in principle.

  The **weak formulations** are:

$$f_i(X(t)) \approx \frac{\mathbb{E}[X_i(t+\Delta t) - X_i(t-\Delta t)]}{2\Delta t}, \tag{3.2.15}$$

$$G_{i,j}(X(t)) \approx \frac{\mathbb{E}[(X_i(t+\Delta t) - X_i(t-\Delta t))(X_j(t+\Delta t) - X_j(t-\Delta t))]}{4\Delta t}. \tag{3.2.16}$$

  The corresponding **strong formulations** are:

$$f_i(X(t)) \approx \frac{X_i(t+\Delta t) - X_i(t-\Delta t)}{2\Delta t}, \tag{3.2.17}$$

$$G_{i,j}(X(t)) \approx \frac{(X_i(t+\Delta t) - X_i(t-\Delta t))(X_j(t+\Delta t) - X_j(t-\Delta t))}{4\Delta t}. \tag{3.2.18}$$

  Just "in principle" as their direct application leads to convergence to Stratonovich SODEs rather than Itô ones [139, Remark 2]. However, the results in Section 3.7 show that they still work correctly, at least for the models considered therein.

- **Trapezoidal rule (TR).** Following the approach of [139], trapezoidal rule offer an alternative to CD to avoid inaccurate convergence behaviour and improve robustness. These schemes combine information from multiple time steps to approximate the integral of the drift and diffusion functions, leveraging implicit integration techniques. This approach is particularly useful when the time discretisation is coarse and provides better stability properties than CD approximations in certain regimes.

  The **weak formulations** are:

$$\frac{f_i(X(t+\Delta t)) + f_i(X(t))}{2} \approx \frac{\mathbb{E}[X_i(t+\Delta t) - X_i(t)]}{\Delta t}, \tag{3.2.19}$$

$$
\begin{aligned}
G_{i,j}(X(t+\Delta t)) + G_{i,j}(X(t)) \approx \mathbb{E}\Bigg[\frac{1}{2\Delta t}\{&[2(X_i(t+\Delta t) - X_i(t)) \\
&-\Delta t(f_i(X(t+\Delta t)) + f_i(X(t)))] \\
&\cdot[2(X_j(t+\Delta t) - X_j(t)) \\
&-\Delta t(f_j(X(t+\Delta t)) + f_j(X(t)))]\}\Bigg].
\end{aligned}
\tag{3.2.20}
$$

The corresponding **strong formulations** are:

$$
\frac{f_i(X(t+\Delta t)) + f_i(X(t))}{2} \approx \frac{X_i(t+\Delta t) - X_i(t)}{\Delta t}, \tag{3.2.21}
$$

$$
\begin{aligned}
G_{i,j}(X(t+\Delta t)) + G_{i,j}(X(t)) \approx \frac{1}{2\Delta t}\{&[2(X_i(t+\Delta t) - X_i(t)) \\
&-\Delta t(f_i(X(t+\Delta t)) + f_i(X(t)))] \\
&\cdot[2(X_j(t+\Delta t) - X_j(t)) \\
&-\Delta t(f_j(X(t+\Delta t)) + f_j(X(t)))]\}.
\end{aligned}
\tag{3.2.22}
$$

All KM, FD, CD, and TR estimates are employed and compared in Section 3.7. As discussed in Section 3.5, resorting to their weak or strong form depends on the specific strategy to be adopted, in view of the available data.

The extension to SDDEs follows straightforwardly for the Itô-Taylor expansions by formally replacing the state variable $X(t) \in \mathbb{R}^n$ of the SODE (3.2.1) with an augmented state variable $Z(t) := (X(t), X(t-\tau)) \in \mathbb{R}^{2n}$ from the SDDE (3.2.2). Regarding the related weak estimates of drift and covariance, we stress that conditioning must include both current and delayed states for consistent parameter estimation owing to the non-Markovian nature of SDDEs. As an instance, (3.2.7) and (3.2.8) become

$$
f_i(z) \approx \mathbb{E}\left[\frac{X_i(t+\Delta t) - X_i(t)}{\Delta t} \mid X(t) = z_1,\ X(t-\tau) = z_2\right],
$$

$$
G_{i,j}(z) \approx \mathbb{E}\left[\frac{(X_i(t+\Delta t) - X_i(t))(X_j(t+\Delta t) - X_j(t))}{2\Delta t} \mid X(t) = z_1,\ X(t-\tau) = z_2\right],
$$

by choosing $\phi(z) = z_{1,i}$ in the Itô-Taylor expansion. Above, the block partition of $z \in \mathbb{R}^{2n}$ is intended as $(z_1, z_2) \in \mathbb{R}^n \times \mathbb{R}^n$ with z_1 representing the current time value and z_2 the delayed counterpart.

3.3 SINDy for stochastic ordinary differential equations

The application of SINDy to SODEs such as (3.2.1) is first considered in [12] and more recently in [139] (see also the references therein and [88, 137]). The main difference from the use of deterministic differential equations, where data $\mathbf{X}'$ are either available or can be easily recovered from $\mathbf{X}$, is the need for reasonable data for drift and diffusion. To understand the principle underlying the extension of SINDy to SODEs, we can assume that such data are reconstructed or estimated in a non-parametric fashion and are then available as $\mathbf{f}(\mathbf{X}) \in \mathbf{R}^{m\times n}$ for drift and $\mathbf{G}(\mathbf{X}) \in \mathbf{R}^{m\times n^2}$ for diffusion in the form of vectorised covariance. It is then enough to choose two possibly different libraries of candidate functions $\Theta_f \in \mathbb{R}^{m\times p_f}$ and $\Theta_G \in \mathbb{R}^{m\times p_G}$ and solving independently

$$\alpha_i = \arg\min_{\alpha\in\mathbb{R}^p} \left(\left\|\mathbf{f}_i(\mathbf{X}) - \Theta_f(\mathbf{X})\alpha\right\|_2 + \lambda_f \|\alpha\|_1 \right), \quad i = 1, \dots, n, \tag{3.3.1}$$

$$\beta_{i,j} = \arg\min_{\beta\in\mathbb{R}^p} \left(\left\|\mathbf{G}_{i,j}(\mathbf{X}) - \Theta_G(\mathbf{X})\beta\right\|_2 + \lambda_G \|\beta\|_1 \right), \quad i, j = 1, \dots, n. \tag{3.3.2}$$

Note the freedom of choosing different sparsity promoting parameters among drift and diffusion. The regression problem is solved using the STLS algorithm as defined in Chapter 1. In Section 3.5 a detailed discussion on how to obtain $\mathbf{f}(\mathbf{X})$ and $\mathbf{G}(\mathbf{X})$ using the three different approaches. First, we deal with SINDy for SDDEs as the primary methodological contribution of the present chapter.

Strictly speaking, due to state-dependent diffusion the data are heteroscedastic and a generalised least-squares formulation with a variance model based on G would be more appropriate. Here, we use ordinary least squares as a first-order approximation; extending the framework to fully account for heteroscedasticity is an interesting direction for future work.

3.4 SINDy for stochastic delay differential equations

To the best of our knowledge, the methodology developed in this chapter represents the first general approach for identifying SDDEs using sparse identification techniques without requiring prior knowledge of functional forms. The existing literature on this topic is limited to the recent work by [64], which proposes an SINDy based method under substantial restrictions. Their approach applies Taylor expansion to eliminate delayed terms from the drift and diffusion coefficients, effectively reducing the problem to a delay free approximation. Although this provides a valuable starting point, the methodology is demonstrated exclusively on systems with small, single time delays, and the accuracy degrades significantly as the delay parameter increases, limiting applicability to small delay regimes. Furthermore, the framework restricts treatment to Stratonovich type stochastic integrals, and multiple delay systems are mentioned only briefly without experimental validation. In contrast, this chapter develops a comprehensive framework capable of handling arbitrary delays, multiple delay structures, and a flexible treatment of stochastic integral types, thus filling an important gap in the current literature.

To address these gaps, we propose a general extension of SINDy to SDDEs by combining two key concepts: the augmented library framework from the SINDy approach for DDEs (incorporating delayed samples) and the covariance-based estimation strategies from Section 3.3 for SODEs.

Indeed, the key observation is that, in principle, the presence of delay terms requires an augmented library that includes delayed samples $\mathbf{X}_\tau$, while the presence of stochastic terms requires suitable strategies to recover data $\mathbf{f}(\mathbf{X})$ for drift and $\mathbf{G}(\mathbf{X})$ for diffusion through covariance.

As such, the two issues appear basically unrelated, and this would lead to extending (3.3.1) and (3.3.2) as follows:

$$\alpha_i = \arg\min_{\alpha \in \mathbb{R}^p} \left(\left\| \mathbf{f}_i(\mathbf{X}) - \Theta_f(\mathbf{X}, \mathbf{X}_\tau)\alpha \right\|_2 + \lambda_f \|\alpha\|_1 \right), \quad i = 1, \ldots, n,$$

$$\beta_{i,j} = \arg\min_{\beta \in \mathbb{R}^p} \left(\left\| \mathbf{G}_{i,j}(\mathbf{X}) - \Theta_G(\mathbf{X}, \mathbf{X}_\tau)\beta \right\|_2 + \lambda_G \|\beta\|_1 \right), \quad i, j = 1, \ldots, n.$$

Nevertheless, a more reliable approach should instead consider the explicit dependence on the delayed samples also for the reconstruction of data for drift and diffusion; hence

$$\alpha_i = \arg\min_{\alpha \in \mathbb{R}^p} \left(\left\| \mathbf{f}_i(\mathbf{X}, \mathbf{X}_\tau) - \Theta_f(\mathbf{X}, \mathbf{X}_\tau)\alpha \right\|_2 + \lambda_f \|\alpha\|_1 \right), \quad i = 1, \ldots, n, \tag{3.4.1}$$

$$\beta_{i,j} = \arg\min_{\beta \in \mathbb{R}^p} \left(\left\| \mathbf{G}_{i,j}(\mathbf{X}, \mathbf{X}_\tau) - \Theta_G(\mathbf{X}, \mathbf{X}_\tau)\beta \right\|_2 + \lambda_G \|\beta\|_1 \right), \quad i, j = 1, \ldots, n. \tag{3.4.2}$$

Indeed, as discussed at the end of Section 3.2 in view of the Itô-Taylor expansions for SDDEs, weak forms generally require conditioning also w.r.t. past values; thus, (3.3.1) and (3.3.2) fit better into the framework. The issue becomes evident in next Section 3.5 regarding one of the possible strategies to recover the necessary data.

3.5 Estimation of drift and diffusion from data

In this section, we obtain the data $\mathbf{f}(\mathbf{X})$ and $\mathbf{G}(\mathbf{X})$ as anticipated in Section 3.4. In Section 3.2 general moment-based estimators are described using Itô-Taylor expansions. Independently of the specific choice or order, the KM, FD, CD, and TR methods all face a fundamental challenge: how to obtain sufficient data at each time instant to ensure reliable estimates of drift and diffusion. The answer depends, among other factors, on whether temporal data are available for a single trajectory alone or samples from multiple independent realisations are at disposal (realistically from repeated experiments). In the first case, the approach we follow, hereafter named "approach A" or simply "A" for brevity, relies on creating more realisations via suitable tools and is described in Section 3.5.1. In the second case, "approach B" or simply "B" in the following, we discuss two different strategies, namely B1 and B2, described in Sections 3.5.2 and 3.5.3. The key difference between B1 and B2 is whether the averages are taken before or after the application of SINDy's sparse regression. In Section 3.7 we perform a thorough experimental comparison of different models of SDDEs, illustrating different trade-offs between reconstruction error, computational cost, data requirements, and implementation complexity.

In the presence of delay an augmented state representation of SDDEs take the form

$$Z(t) := (X(t), X(t-\tau)) \in \mathbb{R}^{2n}, \tag{3.5.1}$$

as discussed at the end of Section 3.2. This augmented formulation enables the standard estimation methods (developed for non-delayed systems) to properly account for temporal dependencies introduced by the delays.

We emphasise that whilst the general properties and effectiveness of these estimation tools have been rigorously established for non-delayed systems, formal theoretical analysis of their performance in the presence of delays remains an open question. Instead, we rely on comprehensive experimental validation in Section 3.7, which provides empirical evidence of the efficacy and trade-offs between reconstruction accuracy, computational cost, data requirements, and implementation complexity across various SDDE models.

3.5.1 Approach A: more realisations from conditional expectations

In this section, we resort to the methodology discussed in [40]. The underlying basis is also used in the first work [12] on SINDy for SODEs by employing KM estimates.

Let δt be the stepsize chosen to represent the Brownian motion W underlying (3.2.2). Assume that samples of a single path $X(t)$ are available with time stepsize $\Delta t = K\delta t$ for some integer $K \geq 1$ (for our simulation, we uses $K = 1$ in all the results presented in Section 3.7). If not, one may consider interpolated samples to obtain delayed samples and hence $Z(t)$. Given the relevant time instants t_i, $i = 1, \ldots, m$ and a positive integer M_{traj}, for each $i = 1, \ldots, m-1$ the proposed methodology first requires the following three steps:

A(i) initialise M_{traj} independent sample paths $X^{(k)}(t)$, $k = 1, \ldots, M_{\text{traj}}$, of (3.2.2) each starting at time t_i from $Z(t_i)$;

A(ii) approximate a single step of each path at time $t_i + \Delta t$ by using a numerical time-integrator, e.g., Euler–Maruyama as adapted to (3.2.2) [32], viz.

$$X_{i+1}^{(k)} = X_i^{(k)} + f(X_i^{(k)}, X_{i-\kappa}^{(k)})\Delta t + g(X_i^{(k)}, X_{i-\kappa}^{(k)})(W^{(k)}(t_{i+1}) - W^{(k)}(t_i)),$$

where we implicitly assume that τ is an integer multiple κ of Δt;

A(iii) record the increment $\Delta X_i^{(k)} := X_{i+1}^{(k)} - X_i^{(k)}$.

Then estimates of drift and diffusion (again through covariance) are obtained as

$$\mathbf{f}(Z(t_i)) = \frac{1}{M_{\text{traj}}\Delta t} \sum_{k=1}^{M_{\text{traj}}} \Delta X_i^{(k)},$$

$$\mathbf{G}(Z(t_i)) = \frac{1}{\Delta t} \left(\frac{1}{M_{\text{traj}}} \sum_{k=1}^{M_{\text{traj}}} \Delta X_i^{(k)} \left(\Delta X_i^{(k)}\right)^T - \mathbf{f}(Z(t_i))\mathbf{f}(Z(t_i))^T(\Delta t)^2 \right).$$

As $M_{\text{traj}} \to \infty$ and $\Delta t \to 0$, these estimates are expected to converge to the true values $\mathbf{f}(Z(t_i))$ and $\mathbf{G}(Z(t_i))$ (which is indeed the case without delay [40]). In practice, one chooses K large enough ($\delta t \ll \Delta t$) to control the discretisation bias and M_{traj} large enough to reduce the sampling variance. Higher-order schemes (e.g. Milstein [99] and stochastic Runge–Kutta [82]) or finite-time corrections [107] can be used when Δt is not negligible.

A final comment is fundamental, also in view of approaches B1 and B2, as described next: the above procedure requires an assumed specification or initial approximation of both f and g; otherwise, step A(ii) is not possible. In practice, these functions are supplied via a suitable parametric family, a non-parametric initialiser, or prior structural information, and then refined iteratively until a desired tolerance [40]. In Section 3.7 we adopt the true forms of f and g to avoid complexities unrelated to the current scope.

In the numerical experiments, Approach A is initialised with the true functional forms f_0 and g_0 so that the focus is on evaluating the moment estimators rather than on model discovery. In this setting, the method effectively recovers noisy versions of the true drift and diffusion but is not the main identification strategy of this thesis. Developing a fully iterative version of Approach A, starting from generic parametric models and analysing its convergence, lies beyond the scope of this work and is left for future research.

3.5.2 Approach B1: multiple trajectories with pre-regression averaging

An alternative to approach A can be followed when a large dataset from several independent realisations of the system is available, as follows:

Consider M_{traj} independent trajectories of (3.2.2), each sampled at t_i, $i = 1, \dots, m$, with a uniform time step Δt. Let $Z^{(k)}(t_i)$ be the augmented state of the k-th trajectory at time t_i and $\Delta X^{(k)}(t_i) = X^{(k)}(t_{i+1}) - X^{(k)}(t_i)$ be the relevant increment of the current time state. To estimate drift and diffusion at a specific point $z \in \mathbb{R}^{2n}$ in the augmented state space, we aggregate data points from all trajectories that fall within a predefined neighbourhood of z [6, 100]. Given a tolerance $\epsilon > 0$ let $N_\epsilon(z)$ denote the cardinality of the set $S_\epsilon(z) := \{(k,i) \ : \ \|Z^{(k)}(t_i) - z\| \leq \epsilon\}$ of such observations. This effectively performs a kernel-like local averaging, where ϵ controls the trade-off between the statistical variance (too few points if ϵ is small) and bias (over-smoothing if ϵ is large). This and similar techniques are widely used in non-parametric drift and diffusion estimation methods for stochastic processes, for instance, through kernel density estimation or binning schemes in continuous-time models. The required estimates are finally obtained by averaging the observed increments and quadratic variations over all data points in the neighbourhood of z, as follows:

$$\mathbf{f}(z) = \frac{1}{N_\epsilon(z)\Delta t} \sum_{k=1}^{M_{\text{traj}}} \sum_{i:(k,i)\in S_\epsilon(z)} \Delta X_{t_i}^{(k)}, \tag{3.5.2}$$

$$\mathbf{G}(z) = \frac{1}{N_\epsilon(z)\Delta t} \sum_{k=1}^{M_{\text{traj}}} \sum_{i:(k,i)\in S_\epsilon(z)} \Delta X_{t_i}^{(k)} (\Delta X_{t_i}^{(k)})^T. \tag{3.5.3}$$

The estimates presented are based on the KM in an ensemble setting. Generalisation to FD, CD, and TR (presented in Section 3.2) is straightforward. Convergence to $\mathbf{f}(z)$ and $\mathbf{G}(z)$ is expected as $M_{\text{traj}} \to \infty$ (or as the total number of observations $N_\epsilon(z)$ increases) for SDDEs. The accuracy depends on balancing the bias (from finite Δt and neighbourhood size ϵ) and variance (from a limited amount of data). As a final comment, we observe that this approach facilitates a sparse, data-driven reconstruction of the underlying non-Markovian stochastic dynamics of SDDEs without requiring explicit knowledge of the functional forms of f and g beforehand. Moreover, we note that B1 is characterised by taking averages before the application of SINDy's sparse regression, contrary to B2, as discussed next.

3.5.3 Approach B2: multiple trajectories with post-regression averaging

The second strategy for leveraging an ensemble of M_{traj} trajectories consists of applying SINDy's sparse regression to each trajectory individually and then averaging the identified model parameters. This method, described in the experimental setup of [139], contrasts with B1 by averaging in the parameter space after regression rather than in the data space before regression. Briefly, if $\alpha_i^{(k)}$ and $\beta_{i,j}^{(k)}$ are the sparse coefficients identified by SINDy from the k-th trajectory through (3.3.1) and (3.3.2), respectively, the final model coefficients are retrieved by averaging across the entire ensemble:

$$\alpha_i = \frac{1}{M_{\text{traj}}} \sum_{k=1}^{M_{\text{traj}}} \alpha_i^{(k)}, \qquad \beta_{i,j} = \frac{1}{M_{\text{traj}}} \sum_{k=1}^{M_{\text{traj}}} \beta_{i,j}^{(k)}.$$

Practically, B2 creates an ensemble of models, with the final model being the average of this ensemble. The underlying assumption is that, while each individually identified model may contain variance owing to the specific realisation of the stochastic process, averaging over many such models yields a result that is expected to converge to the true underlying system dynamics.

The simple averaging of parameter estimates across trajectories assumes linear dependence on the parameters. When parameters are obtained from nonlinear optimisation, Jensen's inequality [70] implies that the average of individual estimates need not coincide with the minimiser of the average loss. This subtlety should be kept in mind when interpreting B2-type averages.

3.6 Algorithmic implementation

The SINDy framework for SDDEs presents three distinctive methodologies, one for each data setting and model identification objective. Detailed step-by-step pseudocode and complete algorithmic descriptions are presented in Appendix A. Algorithm A.4 outlines Approach A (3.5.1), which employs a single available trajectory to generate synthetic trajectories through conditioning of observed states and simulation of stochastic increments under an initially user-provided model. It is marked by efficiency and is thus best suited for initial estimates in cases of partly known model shapes or fast results generation. However, the accuracy of this technique depends on the quality of the initial estimates and does not significantly ease the discoverability of the model in cases with uncertain dynamics.

Approach B1 (3.5.2) uses several independent trajectories to perform local averaging of state increments in certain neighbourhoods prior to sparse regression, as described in Algorithm A.5. Using direct averaging in the data space instead of the parameter space, B1 (3.5.2) exhibits significant robustness, allows nonparametric identification of the model, and always results in higher accuracy in the estimation of diffusion and drift coefficients in reconstruction. It is advisable for whole model structure discovery and verification, particularly in situations wherein there is no prior information, although it requires more computational effort.

Algorithm A.6 performs Approach B2 (3.5.3) and requires multiple trajectories but employs regression only once for each, averaging the subsequent estimated parameters. Conceptually, it better lends itself to ensemble modelling and to estimation of uncertainty, but B2 tends to yield less accurate reconstructions and larger errors; as such, it would not be a leading technique in system identification.

3.7 Numerical experiments and results

In this section, we perform a thorough computational analysis by conducting several experiments on the three SDDEs described below. The experiments consist of applying the approaches A, B1, and B2 discussed in Section 3.5, each of them using all of KM, FD, CD, and TR as underlying estimators from Section 3.2. We recall that M_{traj} denotes both the number of paths realised with A through Euler-Maruyama simulations and the cardinality of the ensemble of available trajectories for B1 and B2. A complete comparison is first performed for all three examples by assuming $M_{\text{traj}} = 1\,000$ and $\epsilon = 10^{-4}$ for B1. Further tests regard limited comparisons to separately assess the role of M_{traj} and ϵ also w.r.t. CPU time and errors. For the latter, we consider both the absolute errors $|\bar{\xi}_i - \xi_i|$ on the i-th sparse coefficient of the chosen SINDy library, $i = 1, \ldots, p$, as well as the RMSE on relevant trajectories (they regard both trajectories for the state $X(t)$ as well as for separated drift and diffusion dynamics reconstruction). Finally, data are artificially provided by numerical simulation via Euler-Maruyama, and we use STLS [31] for sparse regression due to its robustness compared to other methods. Before discussing the results obtained in Section 3.7.1 we present the three specific SDDEs that we use as models to test.

Example 3.7: stochastic delay logistic equation

As a first model we consider from [4, 145] the stochastic version with multiplicative noise

$$\mathrm{d}X(t) = \alpha X(t)\big(1 - X(t-\tau)\big)\,\mathrm{d}t + \sigma X(t)\,\mathrm{d}W(t) \tag{3.7.1}$$

of the so-called delay logistic or Hutchinson equation [69]. (3.7.1) models the evolution of a single ($n = 1$) population $X(t)$ at time t with intrinsic growth rate $\alpha = 2$, time delay $\tau = 1$ and noise intensity $\sigma = 0.4$ with underlying scalar ($q = 1$) Brownian motion. Data are obtained by simulating over the time window $[0, 20]$ with time step $\Delta t = 0.01$ via Euler–Maruyama, prescribing the history $X(s) = \cos(s)$, $s \in [-\tau, 0]$. 80% of the samples were used for training and 20% for validation. SINDy is applied with polynomial libraries of degree $d = 2$ for both drift and diffusion and $\lambda = 0.025$ is used.

The results of the full comparison are presented in Table 3.7.1 and discussed in Section 3.7.1.

	library terms	true coeff.	$\lvert\xi_i - \bar{\xi}_i\rvert$											
			KM			FD			CD			TR		
			A	B1	B2	A	B1	B2	A	B1	B2	A	B1	B2
drift	$X(t)$	$\xi_2 = 2$	$2.13e^{-3}$	$\mathbf{1.29e^{-4}}$	$5.76e^{-1}$	$2.64e^{-2}$	$\mathbf{1.28e^{-4}}$	$5.54e^{-1}$	$1.11e^{-2}$	$\mathbf{3.87e^{-4}}$	$3.36e^{-2}$	$\mathbf{2.13e^{-3}}$	$\mathbf{2.13e^{-3}}$	$1.02e^{-1}$
	$X(t)X(t-\tau)$	$\xi_5 = 2$	$5.07e^{-3}$	$\mathbf{6.68e^{-5}}$	$7.47e^{-2}$	$2.53e^{-2}$	$\mathbf{7.14e^{-5}}$	$6.79e^{-2}$	$1.05e^{-2}$	$\mathbf{2.08e^{-4}}$	$6.57e^{-2}$	$2.10e^{0}$	$\mathbf{1.17e^{-3}}$	$1.60e^{-2}$
same diffusion														
diffusion	$X^2(t)$	$\xi_4 = 0.16$	$6.39e^{-3}$	$\mathbf{7.54e^{-9}}$	$9.17e^{-3}$	$6.39e^{-3}$	$\mathbf{7.54e^{-9}}$	$9.17e^{-3}$	$6.39e^{-3}$	$\mathbf{7.54e^{-9}}$	$9.17e^{-3}$	$6.39e^{-3}$	$\mathbf{7.54e^{-9}}$	$9.17e^{-3}$
performance metrics														
CPU [s]			≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10
RMSE	full		$2.99e^{-1}$	$\mathbf{1.10e^{-3}}$	$8.79e^{-1}$	$3.43e^{-1}$	$\mathbf{4.29e^{-4}}$	$9.80e^{-1}$	$3.31e^{-1}$	$\mathbf{1.30e^{-3}}$	$1.10e^{0}$	$3.73e^{0}$	$\mathbf{1.43e^{-3}}$	$1.68e^{0}$
	drift		$1.11e^{-1}$	$\mathbf{1.13e^{-4}}$	$3.79e^{-1}$	$2.18e^{-2}$	$\mathbf{9.75e^{-5}}$	$4.68e^{-1}$	$9.00e^{-3}$	$\mathbf{3.07e^{-4}}$	$1.37e^{-1}$	$5.47e^{0}$	$\mathbf{1.70e^{-3}}$	$1.13e^{0}$
	diffusion		$9.99e^{-2}$	$\mathbf{4.92e^{-6}}$	$2.43e^{-1}$	$9.97e^{-2}$	$\mathbf{2.85e^{-6}}$	$2.21e^{-1}$	$9.97e^{-2}$	$\mathbf{1.53e^{-5}}$	$1.97e^{-1}$	$9.99e^{-2}$	$\mathbf{1.22e^{-5}}$	$2.20e^{-1}$
different diffusion														
diffusion	$X^2(t)$	$\xi_4 = 0.16$	$6.39e^{-3}$	$\mathbf{7.54e^{-9}}$	$8.31e^{-2}$	$7.91e^{-2}$	$\mathbf{1.50e^{-5}}$	$7.92e^{-2}$	$8.48e^{-2}$	$\mathbf{7.76e^{-3}}$	$9.47e^{-2}$	$1.84e^{-1}$	$\mathbf{2.90e^{-6}}$	$6.95e^{-2}$
performance metrics														
CPU [s]			≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10
RMSE	full		$2.99e^{-1}$	$\mathbf{1.10e^{-3}}$	$9.43e^{-1}$	$5.25e^{-1}$	$\mathbf{6.00e^{-3}}$	$1.08e^{0}$	$6.49e^{-1}$	$\mathbf{2.99e^{-2}}$	$1.89e^{0}$	$2.62e^{0}$	$\mathbf{3.00e^{-2}}$	$9.83e^{-1}$
	drift		$1.11e^{-2}$	$\mathbf{1.13e^{-4}}$	$1.15e^{0}$	$1.14e^{-2}$	$\mathbf{9.06e^{-4}}$	$3.59e^{-1}$	$\mathbf{9.00e^{-3}}$	$2.32e^{-2}$	$1.56e^{-1}$	$5.47e^{0}$	$\mathbf{1.10e^{-3}}$	$2.90e^{-1}$
	diffusion		$9.99e^{-2}$	$\mathbf{4.92e^{-6}}$	$1.57e^{0}$	$3.08e^{-1}$	$\mathbf{3.89e^{-5}}$	$1.05e^{0}$	$2.69e^{-1}$	$\mathbf{1.54e^{-2}}$	$8.97e^{-1}$	$6.21e^{-1}$	$\mathbf{7.64e^{-6}}$	$2.41e^{-1}$

Table 3.7.1: comparison of errors on estimated coefficients, RMSE on trajectories and CPU time for the logistic SDDE (3.7.1) for both drift and diffusion, all estimation methods KM, FD, CD and TR from Section 3.2 and all data reconstruction strategies A, B1 and B2 from Section 3.5. Diffusion estimation might remain the same (KM in particular, first part above) or change (all of KM, FD, CD, and TR, second part below).

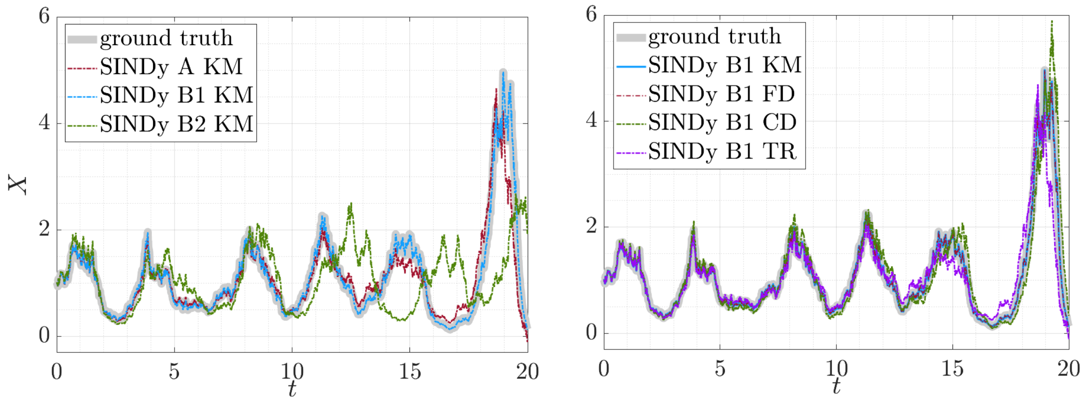

Figure 3.7.1: comparison of trajectories for the logistic SDDE (3.7.1): ground truth and approaches A, B1 and B2 all with KM (left). Ground truth versus estimation methods KM, FD, CD and TR all with approach B1 (right) [18].

Example 3.7: stochastic predator prey model

As a second model we consider from [49] the predator–prey interaction with delay and multiplicative noise

$$\begin{cases} \mathrm{d}X_1(t) = X_1(t)[\alpha - \beta X_1(t) - \gamma X_2(t-\tau)]\,\mathrm{d}t + \sigma_1 X_1(t)\,\mathrm{d}W_1(t) \\ \mathrm{d}X_2(t) = X_2(t)[-\delta + \kappa X_1(t-\tau)]\,\mathrm{d}t + \sigma_2 X_2(t)\,\mathrm{d}W_2(t), \end{cases} \tag{3.7.2}$$

where $X_1(t)$ and $X_2(t)$ denote prey and predator populations, respectively, with parameters $\alpha = 1$, $\delta = 0.5$, $\beta = \gamma = \kappa = 0.1$, $\sigma_1 = \sigma_2 = 0.4$, $\tau = 1$ and W_1, W_2 independent scalar Wiener processes (here $n = q = 2$). Again, data are obtained by simulating over the time window $[0, 20]$ with time step $\Delta t = 0.01$ via Euler–Maruyama, prescribing the constant history $X(s) \equiv (5, 2)^T$. 80% of the samples were used for training and 20% for validation. SINDy is applied with polynomial libraries of degree $d = 2$ for both drift and diffusion and $\lambda = 0.025$ is used.

The results of the full comparison are presented in Table 3.7.2 and discussed in Section 3.7.1.

	library terms	true coeff.	KM A	KM B1	KM B2	FD A	FD B1	FD B2	CD A	CD B1	CD B2	TR A	TR B1	TR B2
			$\vert\xi_i - \bar{\xi}_i\vert$											
drift	$X(t)$	$\xi_2 = 1$	$1.28e^{-1}$	$\mathbf{9.20e^{-5}}$	$2.27e^{0}$	$6.90e^{-2}$	$\mathbf{5.19e^{-4}}$	$2.26e^{0}$	$1.16e^{-2}$	$\mathbf{5.52e^{-4}}$	$1.80e^{-1}$	$3.23e^{-1}$	$\mathbf{1.85e^{-5}}$	$4.51e^{0}$
	$Y(t)$	$\xi_3 = 0.5$	$1.42e^{-3}$	$\mathbf{2.86e^{-7}}$	$1.13e^{0}$	$7.14e^{-2}$	$\mathbf{2.32e^{-4}}$	$1.55e^{0}$	$1.78e^{-2}$	$\mathbf{2.56e^{-4}}$	$3.60e^{-1}$	$7.86e^{-3}$	$\mathbf{1.38e^{-6}}$	$15.58e^{0}$
	$X^2(t)$	$\xi_6 = 0.1$	$6.16e^{-3}$	$\mathbf{8.38e^{-6}}$	$1.65e^{-1}$	$6.59e^{-3}$	$\mathbf{3.98e^{-5}}$	$1.73e^{-1}$	$1.64e^{-3}$	$\mathbf{5.09e^{-5}}$	$7.88e^{-2}$	$1.68e^{-2}$	$\mathbf{5.06e^{-5}}$	$6.33e^{-1}$
	$X(t)Y(t-\tau)$	$\xi_9 = 0.1$	$1.05e^{-2}$	$\mathbf{3.07e^{-6}}$	$1.42e^{-1}$	$6.55e^{-3}$	$\mathbf{5.86e^{-5}}$	$1.45e^{-1}$	$7.63e^{-4}$	$\mathbf{1.60e^{-5}}$	$2.20e^{-2}$	$1.36e^{-1}$	$\mathbf{3.79e^{-5}}$	$5.60e^{-1}$
	$Y(t)X(t-\tau)$	$\xi_{11} = 0.1$	$7.35e^{-4}$	$\mathbf{1.84e^{-8}}$	$5.07e^{-2}$	$1.08e^{-2}$	$\mathbf{2.35e^{-5}}$	$6.35e^{-2}$	$3.07e^{-3}$	$\mathbf{2.77e^{-5}}$	$1.15e^{-2}$	$1.30e^{-2}$	$\mathbf{1.45e^{-7}}$	$2.46e^{-1}$
same diffusion														
diffusion	$X^2(t)$	$\xi_6 = 0.16$	$6.36e^{-4}$	$\mathbf{2.34e^{-6}}$	$1.10e^{-2}$	$6.36e^{-4}$	$\mathbf{2.34e^{-6}}$	$1.10e^{-2}$	$6.36e^{-4}$	$\mathbf{2.34e^{-6}}$	$1.10e^{-2}$	$6.36e^{-4}$	$\mathbf{2.34e^{-6}}$	$1.10e^{-2}$
	$Y^2(t)$	$\xi_{10} = 0.16$	$1.16e^{-4}$	$\mathbf{1.06e^{-9}}$	$3.55e^{-3}$	$1.16e^{-4}$	$\mathbf{1.06e^{-9}}$	$3.55e^{-3}$	$1.16e^{-4}$	$\mathbf{1.06e^{-9}}$	$3.55e^{-3}$	$1.16e^{-4}$	$\mathbf{1.06e^{-9}}$	$3.55e^{-3}$
performance metrics														
CPU [s]			≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30
RMSE		full	$1.07e^{-1}$	$\mathbf{1.00e^{-3}}$	$2.89e^{0}$	$4.84e^{-1}$	$\mathbf{4.80e^{-3}}$	$2.13e^{0}$	$1.50e^{-1}$	$\mathbf{7.20e^{-3}}$	$4.30e^{0}$	$1.70e^{0}$	$\mathbf{5.37e^{-2}}$	$5.10e^{0}$
		drift	$3.10e^{-2}$	$\mathbf{8.80e^{-5}}$	$2.66e^{0}$	$1.00e^{-1}$	$\mathbf{5.40e^{-4}}$	$8.89e^{0}$	$2.63e^{-2}$	$\mathbf{7.50e^{-4}}$	$1.27e^{0}$	$1.39e^{0}$	$\mathbf{4.66e^{-4}}$	$2.03e^{1}$
		diffusion	$2.06e^{-2}$	$\mathbf{7.57e^{-6}}$	$1.59e^{-1}$	$2.28e^{-2}$	$\mathbf{6.62e^{-6}}$	$8.87e^{-2}$	$2.28e^{-2}$	$\mathbf{1.11e^{-5}}$	$6.93e^{-2}$	$2.60e^{-2}$	$\mathbf{6.68e^{-5}}$	$1.41e^{-1}$
different diffusion														
diffusion	$X^2(t)$	$\xi_6 = 0.16$	$6.36e^{-4}$	$\mathbf{2.34e^{-6}}$	$1.10e^{-2}$	$7.91e^{-2}$	$\mathbf{9.35e^{-6}}$	$7.13e^{-2}$	$7.96e^{-2}$	$\mathbf{3.13e^{-3}}$	$1.04e^{-1}$	$1.73e^{-3}$	$\mathbf{1.30e^{-8}}$	$1.10e^{0}$
	$Y^2(t)$	$\xi_{10} = 0.16$	$1.16e^{-4}$	$\mathbf{1.06e^{-9}}$	$3.55e^{-3}$	$8.06e^{-2}$	$\mathbf{2.32e^{-8}}$	$7.76e^{-2}$	$7.97e^{-2}$	$\mathbf{2.65e^{-4}}$	$1.05e^{-1}$	$7.61e^{-5}$	$\mathbf{4.49e^{-9}}$	$7.24e^{-1}$
performance metrics														
CPU [s]			≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30	≈ 5	≈ 30	≈ 30
RMSE		full	$1.07e^{-1}$	$\mathbf{1.00e^{-3}}$	$2.89e^{0}$	$1.17e^{0}$	$\mathbf{2.70e^{-3}}$	$2.21e^{0}$	$1.27e^{0}$	$\mathbf{1.70e^{-2}}$	$1.14e^{0}$	$1.70e^{0}$	$\mathbf{5.80e^{-2}}$	$6.58e^{1}$
		drift	$3.10e^{-2}$	$\mathbf{8.80e^{-5}}$	$2.66e^{0}$	$1.00e^{-1}$	$\mathbf{5.76e^{-4}}$	$8.71e^{0}$	$2.63e^{-2}$	$\mathbf{2.34e^{-2}}$	$1.16e^{0}$	$1.39e^{0}$	$\mathbf{3.90e^{-4}}$	$6.53e^{1}$
		diffusion	$2.06e^{-2}$	$\mathbf{7.57e^{-6}}$	$1.59e^{-1}$	$4.24e^{0}$	$\mathbf{3.38e^{-4}}$	$7.61e^{-1}$	$4.22e^{0}$	$\mathbf{6.06e^{-2}}$	$6.72e^{-1}$	$5.42e^{-2}$	$\mathbf{1.55e^{-6}}$	$1.87e^{0}$

Table 3.7.2: comparison of errors on estimated coefficients, RMSE on trajectories and CPU time for the predator-prey SDDE (3.7.2) for both drift and diffusion, all estimation methods KM, FD, CD and TR from Section 3.2 and all data reconstruction strategies A, B1 and B2 from Section 3.5. Diffusion estimation might remain the same (KM in particular, first part above) or change (all of KM, FD, CD and TR, second part below).

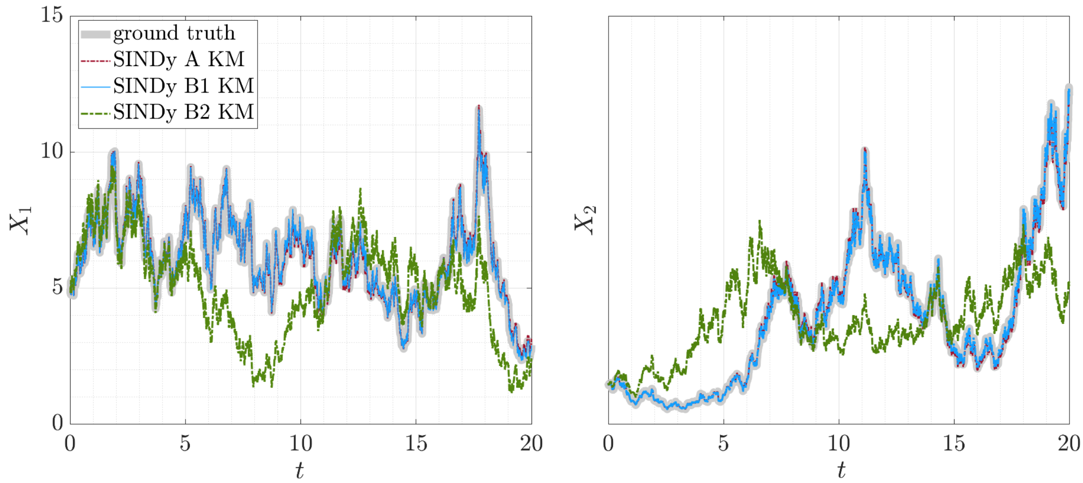

Figure 3.7.2: comparison of trajectories for the predator-prey SDDE (3.7.2): ground truth (solid thick grey) and approaches A (dashed red), B1 (dashed blue) and B2 (dashed green) all with KM [18].

Example 3.7: pricing model with delayed response

Finally, as a third and last model we consider from [77] the SDDE

$$\mathrm{d}X(t) = rX(t)\,\mathrm{d}t + \sigma(X(t), X(t-\tau))X(t)\,\mathrm{d}W(t) \tag{3.7.3}$$

modelling the price dynamics of a financial asset with memory, where $X(t)$ denotes the asset (or option) price at time t, $r = 0.05$ is the constant risk-free interest rate, $\tau = 0.002$, the function σ represents the volatility (see below), which is a continuous function of also the history of the stock price and $W(t)$ is a standard Wiener process. (3.7.3) generalises the classical Black–Scholes framework by incorporating historical dependence into the volatility term, capturing memory effects observed in real markets. Once more, data are obtained by simulating over the time window $[0, 20]$ with a time step $\Delta t = 0.01$ via Euler–Maruyama, prescribing the constant history $X(s) \equiv 100$. 80% of the samples were used for training and 20% for validation.

Custom libraries were constructed for SINDy identification. The drift is identified using a polynomial library of degree $d = 1$ and the diffusion is identified using a polynomial library of degree $d = 4$. Both libraries are augmented with the logarithmic term $\ln(X(t)/X(t-\tau))$ (see (3.7.4) below), and $\lambda = 0.025$ is used. The results of the full comparison are presented in Table 3.7.3 and discussed in Section 3.7.1.

As proposed in [77], the volatility function σ is based on a continuous-time analogue of the GARCH model,

$$\sigma^2(X(t), X(t-\tau)) = \frac{\gamma V}{\alpha + \gamma} + \frac{\alpha}{\tau(\alpha + \gamma)} \ln^2 \left(\frac{X(t)}{X(t-\tau)} \right), \tag{3.7.4}$$

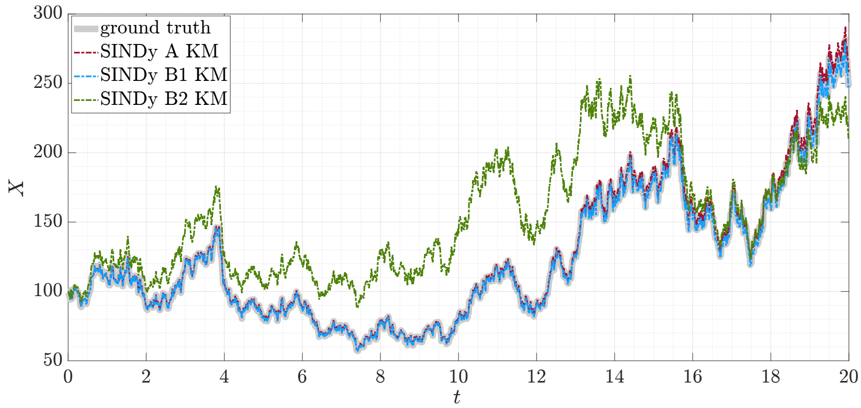

Figure 3.7.3: comparison of trajectories for the option pricing SDDE (3.7.3): ground truth (solid thick grey) and approaches A (dashed red), B1 (dashed blue) and B2 (dashed green) all with KM [18].

where $V = 0.127$ is the long-run average variance, $\alpha = 0.6$ and $\gamma = 0.4$ are positive model parameters controlling the weighting of past returns and reversion strength. This formulation ensures that σ remains strictly positive and reflects the volatility clustering effects observed in empirical financial time series. Such memory-aware volatility plays a crucial role in capturing more realistic option price dynamics than models assuming constant or purely time-dependent volatility.

In this model, the delay τ and the functional structure (including the logarithmic term) are considered constant and assumed known. The algorithm does not estimate the individual parameters V, α, and γ; rather, it identifies their lumped coefficients: the interest rate r for the drift, and the grouped constants $\frac{\gamma V}{\alpha+\gamma}$ and $\frac{\alpha}{\tau(\alpha+\gamma)}$ for the diffusion terms.

3.7.1 Discussion

The analysis of the results collected in Tables 3.7.1, 3.7.2 and 3.7.3 reveals that the optimal estimation strategy is highly dependent on the specific model, the performance metric of interest (error on the reconstructed coefficients, RMSE on trajectories or CPU time) and the chosen methodology (A, B1 or B2 with KM, FD, CD or TR). In general, the pre-regression averaging strategy of B1 outperforms the other choices in terms of accuracy for all three considered models. In contrast, B2 consistently yields the highest absolute errors for every estimated coefficient and the highest RMSE values for the full system as well as for separated drift and diffusion, and given the poor results, it is not considered a viable method. As far as A is concerned, the potential accuracy seems to depend on the prior available knowledge concerning the data via the (one-step) generation of relevant realisations. On the other hand, B1 operates without structural constraints and frequently demonstrates a superior ability to identify the accurate values of individual coefficients, thus highlighting its power in pure parameter and model discovery.

Figures 3.7.1-3.7.3 present a comparison between the ground truth and SINDy predicted trajectories for each model. Figure 3.7.1 (left) compares different estimation approaches for a fixed KM, whereas Figure 3.7.1 (right) compares various methods for esti-

	library terms	true coeff.	$\lvert\xi_i - \bar{\xi}_i\rvert$											
			KM			FD			CD			TR		
			A	B1	B2	A	B1	B2	A	B1	B2	A	B1	B2
drift	$X(t)$	$\xi_2 = 0.05$	$1.49e^{0}$	$\mathbf{4.26e^{-6}}$	$1.32e^{0}$	$1.48e^{0}$	$\mathbf{3.65e^{-6}}$	$1.32e^{0}$	$3.03e^{-1}$	$\mathbf{6.26e^{-6}}$	$1.32e^{0}$	$3.09e^{0}$	$\mathbf{7.57e^{-5}}$	$1.32e^{0}$
same diffusion														
diffusion	$X^2(t)$	$\xi_5 = 0.058$	$5.70e^{-1}$	$\mathbf{2.35e^{-5}}$	$9.20e^{-1}$	$5.70e^{-1}$	$\mathbf{2.35e^{-5}}$	$9.20e^{-1}$	$5.70e^{-1}$	$\mathbf{2.35e^{-5}}$	$9.20e^{-1}$	$5.70e^{-1}$	$\mathbf{2.35e^{-5}}$	$9.20e^{-1}$
	$G(Z(t))$	$\xi_{33} = 300$	$8.51e^{2}$	$\mathbf{1.48e^{0}}$	$1.46e^{2}$	$8.51e^{1}$	$\mathbf{1.48e^{0}}$	$1.46e^{2}$	$8.51e^{1}$	$\mathbf{1.48e^{0}}$	$1.46e^{2}$	$8.51e^{1}$	$\mathbf{1.48e^{0}}$	$1.46e^{2}$
performance metrics														
CPU [s]			≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10
RMSE	full		$3.00e^{0}$	$\mathbf{6.48e^{-6}}$	$3.72e^{0}$	$1.23e^{0}$	$\mathbf{9.67e^{-6}}$	$4.37e^{0}$	$1.12e^{0}$	$\mathbf{6.19e^{-6}}$	$4.42e^{0}$	$7.88e^{0}$	$\mathbf{6.46e^{-3}}$	$2.39e^{1}$
	drift		$5.35e^{-1}$	$\mathbf{2.85e^{-5}}$	$2.50e^{0}$	$7.72e^{-1}$	$\mathbf{1.05e^{-5}}$	$2.32e^{0}$	$2.97e^{-1}$	$\mathbf{7.83e^{-5}}$	$2.48e^{0}$	$1.07e^{0}$	$\mathbf{2.14e^{-4}}$	$9.68e^{1}$
	diffusion		$9.00e^{0}$	$\mathbf{1.17e^{-3}}$	$2.67e^{0}$	$1.13e^{0}$	$\mathbf{4.66e^{-3}}$	$6.69e^{-1}$	$1.13e^{0}$	$\mathbf{1.53e^{-3}}$	$1.29e^{0}$	$9.00e^{0}$	$\mathbf{2.11e^{-2}}$	$1.54e^{0}$
different diffusion														
diffusion	$X^2(t)$	$\xi_5 = 0.058$	$5.70e^{-1}$	$\mathbf{2.35e^{-5}}$	$9.20e^{-1}$	$3.08e^{0}$	$\mathbf{7.37e^{-7}}$	$2.50e^{-1}$	$7.97e^{-1}$	$\mathbf{3.75e^{-6}}$	$5.67e^{-1}$	$6.16e^{-1}$	$\mathbf{1.54e^{-5}}$	$4.82e^{0}$
	$G(Z(t))$	$\xi_{33} = 300$	$8.51e^{2}$	$\mathbf{1.48e^{0}}$	$1.46e^{2}$	$2.93e^{2}$	$\mathbf{7.59e^{-1}}$	$7.46e^{1}$	$2.94e^{2}$	$\mathbf{5.43e^{-1}}$	$2.46e^{2}$	$8.85e^{2}$	$\mathbf{1.40e^{0}}$	$2.90e^{2}$
performance metrics														
CPU [s]			≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10	≈ 4	≈ 10	≈ 10
RMSE	full		$3.00e^{0}$	$\mathbf{6.48e^{-6}}$	$3.72e^{0}$	$3.67e^{1}$	$\mathbf{5.15e^{-6}}$	$4.24e^{1}$	$2.63e^{0}$	$\mathbf{1.03e^{-6}}$	$2.91e^{0}$	$7.76e^{0}$	$\mathbf{4.40e^{-2}}$	$2.76e^{1}$
	drift		$5.35e^{-1}$	$\mathbf{2.85e^{-5}}$	$2.50e^{0}$	$5.34e^{-1}$	$\mathbf{5.36e^{-4}}$	$5.88e^{0}$	$2.97e^{-1}$	$\mathbf{5.27e^{-5}}$	$2.28e^{0}$	$1.07e^{0}$	$\mathbf{7.83e^{-6}}$	$9.13e^{1}$
	diffusion		$9.00e^{0}$	$\mathbf{1.17e^{-3}}$	$2.67e^{0}$	$1.61e^{0}$	$\mathbf{1.58e^{-5}}$	$1.25e^{0}$	$6.47e^{1}$	$\mathbf{1.88e^{-4}}$	$4.43e^{0}$	$9.30e^{0}$	$\mathbf{4.15e^{-3}}$	$1.13e^{0}$

Table 3.7.3: comparison of errors on estimated coefficients, RMSE on trajectories and CPU time for the option-pricing SDDE (3.7.3) with (3.7.4) for both drift and diffusion, all estimation methods KM, FD, CD and TR from Section (3.2) and all data reconstruction strategies A, B1 and B2 from Section 3.5. Diffusion estimation might remain the same (KM in particular, first part above) or change (all of KM, FD, CD and TR, second part below). Recall that $G(Z(t)) = X^2(t)\ln^2(X(t)/X(t-\tau))$.

mation, holding method B1 constant. Based on similar results for the other models, the KM approach was used for all subsequent figures (3.7.2 and 3.7.3).

Figures 3.7.4-3.7.6 provide further clarification on the relative performance of the estimation approaches and numerical schemes under various scenarios. Figure 3.7.4 provides an overall comparison of the RMSE of the state reconstructions for all methodologies as functions of the estimator (KM, FD, CD, TR) and data reconstruction methods (A, B1, B2). This verifies the dominant superiority of the pre-regression averaging scheme B1 independent of the choice of the estimator, with significantly lower RMSE values indicating better reconstruction accuracy. In contrast, B2 consistently provides higher RMSE values, supporting its inappropriateness for trustworthy system identification.

Figures 3.7.5 and 3.7.6 delve deeper into the important methodological parameters affecting the accuracy and computational cost, providing further direction for practical implementation. Figure 3.7.5 examines the effect of the neighbourhood tolerance parameter ϵ in B1 on the absolute errors of the estimated coefficients and CPU time. This insight underscores the necessity of balancing statistical variance and bias during neighbourhood selection to optimise performance. We observe that reducing ϵ too much does not lead to significant improvements and better results are obtained for diffusion rather than drift. Figure 3.7.6 evaluates the impact of ensemble size M_{traj} on estimation accuracy restricted only to the logistic SDDE model (3.7.1) (the results are similar for the other models). We observe that as M_{traj} increases in size, Approach A does not significantly improve the estimation of the diffusion coefficient; however, it leads to better results for the drift term, but only for a substantial increase in M_{traj}, underlying the data-hungry nature of this approach. In contrast, Approach B2 consistently performed poorly for all M_{traj} values. In contrast, Approach B1 consistently outperforms the others in all cases, although its computational cost increases with increasing M_{traj} (larger values of M_{traj} are not considered).

Collectively, these results confirm the numerical advantage of B1 during SDDE model identification in an empirical setting with an unknown model structure. They provide practical advice for hyperparameter optimisation, such as neighbourhood size and cardinality of the ensemble, and thus enable the practical and efficient realisation of the proposed SINDy framework for stochastic delay differential systems. These results parallel and support the overall recommendation to prefer B1 in empirical discovery situations with uncertain model structures.

In short, the selection of an optimal technique depends on the particular analytical goal. If the underlying model structure is specified in advance and one wants to obtain a rapid, inexpensive determination of the aggregate system behaviour, then approach A 3.5.1 offers an appropriate option. However, in a more realistic research application in which the functional shapes of both drift and diffusion terms are still doubtful and must be estimated from observations, approach B1 3.5.2 becomes the preferable option. Its potential to allow system identification without prior assumptions makes it a more robust and scientifically justifiable technique for discovery. While the computational cost is higher, it is justified by its potential to derive the model directly from observation, thus minimising bias imposed by the user and providing more accurate and credible estimations of parameters.

Table 3.7.4 reports the empirical variances of the drift and diffusion coefficients iden-

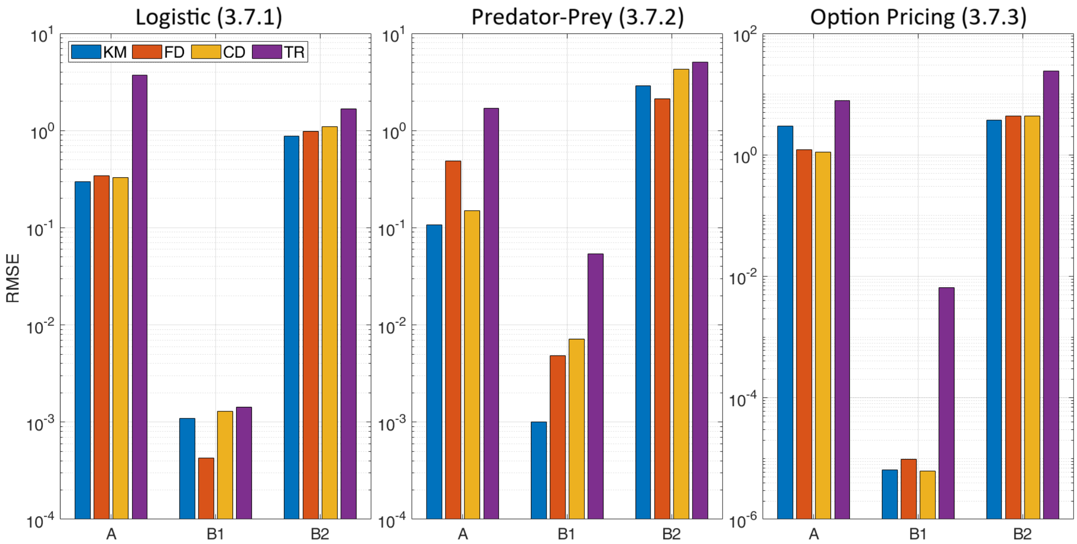

Figure 3.7.4: comparison of RMSE for the SDDEs (3.7.1), (3.7.2) and (3.7.3) with all the approaches A, B1 and B2 against all the estimators KM, FD, CD and TR [18].

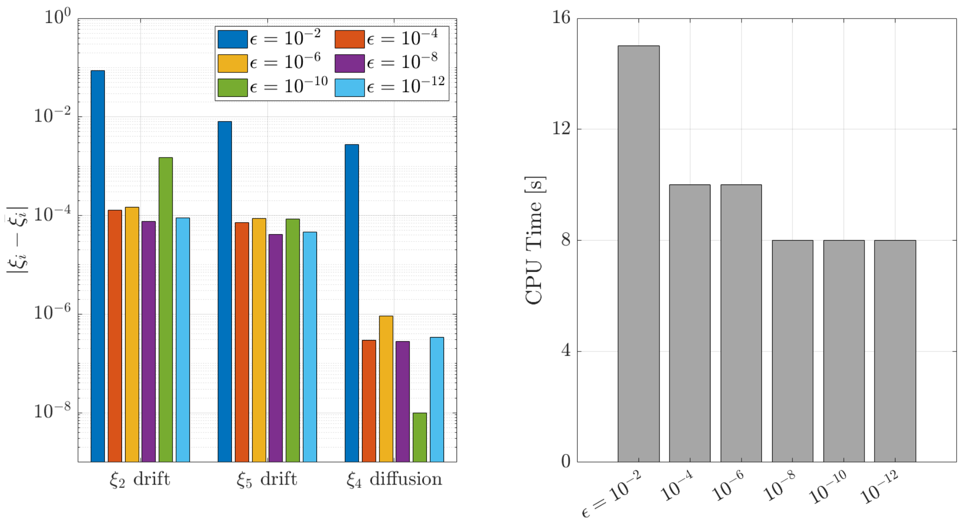

Figure 3.7.5: comparison of the absolute errors $|\xi_i - \bar{\xi}_i|$ (left) and CPU time (right) for varying ϵ for the logistic SDDE (3.7.1) with B1-FD [18].

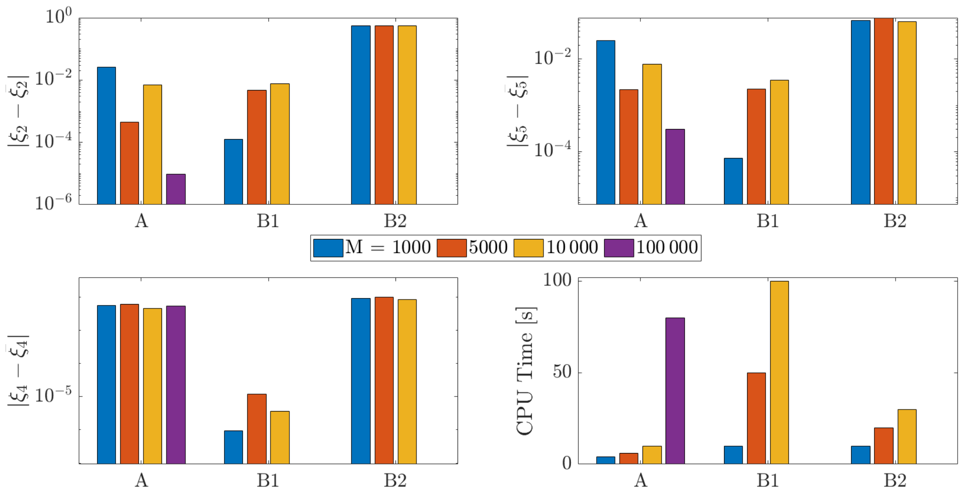

Figure 3.7.6: comparison of the absolute errors $|\xi_i - \bar{\xi}_i|$ for drift (top row) and diffusion (bottom left) for increasing M_{traj} for the logistic SDDE (3.7.1) using A, B1 and B2 all with FD [18].

M_{traj}	A		B1		B2	
	Var drift	Var diff	Var drift	Var diff	Var drift	Var diff
10	$7.78e^1$	$1.01e^0$	$2.91e^1$	$3.25e^{-2}$	$7.76e^1$	$5.03e^{-1}$
100	$7.60e^1$	$9.33e^{-1}$	$8.50e^{-1}$	$2.09e^{-3}$	$3.53e^1$	$3.81e^{-1}$
1000	$7.57e^1$	$9.30e^{-1}$	$8.34e^{-1}$	$1.32e^{-2}$	$3.84e^1$	$4.23e^{-1}$

Table 3.7.4: Empirical variance of the estimated drift and diffusion coefficients for the logistic SDDE (3.7.1) under approaches A, B1 and B2 as a function of the number of trajectories M_{traj}. For each setting, the variance is computed over repeated Monte Carlo runs of the identification procedure.

tified in the logistic SDDE example when approaches A, B1 and B2 are applied to ensembles with $M_{\text{traj}} \in \{10, 100, 1000\}$ independent trajectories. In each case, the variance is computed across Monte Carlo repetitions of the full identification pipeline, and thus quantifies the dispersion of the parameter estimators around their mean, complementing the bias information already shown in Tables 3.7.1 to 3.7.3. Consistent with the error analysis, B1 yields the smallest variances, especially for the diffusion term, whereas B2 exhibits substantially larger variability, confirming that its parameter estimates are statistically much less stable despite being based on the same underlying data. Moreover, the variances for approach A decrease only marginally as M_{traj} increases, which illustrates that this strategy is predominantly limited by the assumptions made in the synthetic resimulation step rather than by the ensemble size.

3.8 Conclusions

This chapter presents a comprehensive framework for sparse identification of SDDEs through the extension of the SINDy methodology. The work addresses the fundamental challenge of simultaneously discovering both drift and diffusion dynamics in systems characterised by non-Markovian behaviour arising from time delays and stochastic perturbations.

The main contributions of this chapter are:

- Approach A, which generates synthetic realisations from a single observed trajectory through conditional sampling, demonstrates intermediate performance that depends critically on the quality of prior knowledge regarding system structure.
- Approach B1, which employs multiple independent trajectories with pre-regression averaging, demonstrates superiority across all tested models.
- Approach B2, which applies sparse regression to individual trajectories before averaging the identified parameters, consistently yields the poorest reconstruction quality.

The investigation yields several significant findings:

- The comparative analysis of moment estimators (KM, FD, CD, TR) reveals that estimator selection plays a secondary role compared to the choice of ensemble averaging strategy. Whilst higher-order estimators (FD, CD, TR) theoretically offer improved bias characteristics, their practical advantage manifests primarily when combined with B1. The KM estimator, when paired with B1, achieves performance comparable to higher-order schemes whilst maintaining computational simplicity.
- The analysis of the tolerance parameter ϵ in approach B1 demonstrates a clear trade-off between statistical variance and bias. Excessively small values of ϵ fail to yield improvement and may increase computational cost without the corresponding accuracy gains.
- Investigation of ensemble cardinality M_{traj} reveals that approach B1 exhibits robust performance with moderate ensemble sizes, whereas approach A requires substantially larger ensembles to achieve comparable accuracy.

3.8.1 Limitations and future directions

Although the proposed framework presented in this chapter represents a significant advance in SDDE identification, several limitations remain:

(*i*) the current methodology assumes that delay values are known a priori or can be estimated independently.

(*ii*) the demonstrated superiority of approach B1 necessitates access to multiple independent realisations of the system.

(*iii*) the curse of dimensionality poses practical challenges for systems.

(*iv*) the accuracy of all approaches critically depends on the temporal resolution of the observations relative to the characteristic timescales of the system.

The combination of delay and stochasticity, while practically effective, remains theoretically incomplete. The augmented state approximation allows standard estimators developed for SODEs to be applied to the delay setting. This is a simplification, because delay equations are inherently infinite-dimensional and non-Markovian, meaning the evolution of the system depends on its continuous history, not just isolated past points. Consequently, conditioning on a finite augmented state lacks formal convergence guarantees. A thorough analysis of how delay and stochasticity interact remains a critical direction for future theoretical research.

Prospectively, the structure offers potential future expansion directions, such as the integration of increasingly complex and problem-specific candidate libraries to tackle more subtle nonlinear and nonlocal dynamics. Furthermore, it must be put into practice and tested on actual empirical data to withstand practical issues such as measurement noise and missing data, in conjunction with fostering hybrid estimation schemes to merge the best properties of both statistical and machine learning formulations. More importantly, structure-preserving machine learning formulations, such as physics-informed neural networks, can be integrated to considerably improve the model stability and interpretability by preserving critical physical constraints. Overall, these directions have potential to facilitate broader applicability and increased impact of sparse identification schemes for SDDEs, to provide robust, interpretable, and computationally tractable data-driven investigations in complex stochastic systems involving memory.

The subsequent chapter extends the SINDy framework to distributed delay systems, addressing the identification of nonautonomous kernel functions that capture the cumulative influence of state histories over continuous delay intervals.

.

4

SINDy for structured populations with distributed memory effects

Contents

4.1 Introduction

The previous chapters have developed sparse identification methods for deterministic and stochastic DDEs, establishing frameworks for discovering governing equations with discrete time delays from observational data. Although these approaches handle systems with one or more discrete delay values, many real-world systems exhibit distributed memory effects [3, 9, 21, 98], where the time evolution depends not on isolated past values but rather on a continuous weighted history of the system. This chapter extends the SINDy framework to distributed delay systems, addressing the challenge of identifying the functional form of distributed memory kernels directly from time-series data. This chapter is primarily based on the submitted work [28].

Delay integro-differential equations (DIDEs) and renewal equations (REs) are fundamental classes of infinite-dimensional dynamical systems that capture systems with distributed memory. These models arise naturally in ecology, epidemiology, physiology [7, 33, 34, 148], and other fields where population structure or system memory depends on continuous distributions of past states rather than discrete delay points. The governing equations feature integral functionals with nonautonomous kernel functions that depend explicitly on both the integration variable (representing age, maturation time, or history) and the state values at past times. Identifying such systems from data presents a difficult problem, reconstructing both the analytical form of kernel and the integration bounds becomes significantly more challenging than handling discrete delays because the number of potential functional forms is uncountably infinite.

A significant gap remains in applying SINDy to systems with distributed memory, which more realistically capture past dependence. Moreover, standard SINDy applications fail to capture the interpretable structure of distributed memory kernels, instead producing high-dimensional black-box models. A robust, data-driven approach is therefore needed that can discover both the analytical structure and parameters of distributed delay systems without relying on extensive prior knowledge.

4.1.1 Problem statement

Identifying DIDEs and REs from observational data presents distinct challenges compared to discrete delay identification. The main issue is discovering g rather than G (the integral of g), as a standard SINDy application would. As a consequence, the nonautonomous nature of g brings in further challenges. Standard SINDy libraries require modifications to capture this structure. Additionally, the integration bounds are often unknown, and the continuous nature of distributed delays leads to a dimensionality problem.

The specific challenges addressed in this chapter include:

- identification of nonautonomous kernel functions from time-series data;
- recovery of unknown integration bounds and other non-multiplicative parameters;
- computational efficiency in high-dimensional parameter space exploration;
- robustness to noise and sampling strategies whilst maintaining interpretability.

4.1.2 Background and related work

Distributed delay systems in applications

Systems with distributed memory have been extensively studied, and as an example, the logistic Daphnia model [22, 26] exemplifies consumer-resource dynamics with age-structured maturation delays. Behavioural epidemiology models [7, 33, 34] incorporate vaccination responses based on the historical prevalence of the disease, weighted through memory kernels. Ricker-type population models [9, 75] employ gamma-distributed maturation delays to describe reproduction dynamics. These applications demonstrate that distributed delays arise naturally and require interpretable identification methods to reveal underlying mechanisms.

State of the art for distributed delay identification

Previous studies have developed theoretical methods for analysing systems with distributed delays based on first-principles approaches, including spectral methods, Floquet theory, and bifurcation analysis with specified kernels. However, data-driven discovery of distributed delay systems remains largely unexplored. The integral SINDy (ISINDy) framework [96, 117, 140], as they are based on the integral formulation of ODEs, provides a foundation by replacing numerical differentiation with integration. This approach shows promise for handling integral functionals because integration is inherently more stable than differentiation in the presence of measurement noise. However, existing integral SINDy methods do not address the nonautonomous nature of distributed kernels. As a result, the extension to DIDEs and REs requires substantial methodological innovation.

Gaps in existing methods

Despite advances in data-driven system identification and theoretical analysis of distributed delay systems, critical gaps remain:

- traditional first-principles approaches require deep domain expertise;
- no systematic data-driven methods exist that preserve interpretability for nonautonomous kernels;
- standard SINDy produces black-box models unsuitable for understanding kernel structure and biological interpretation;
- data-driven discovery of nonautonomous kernels lacks systematic approaches that maintain interpretability;
- ISINDy lacks systematic approaches for discovering nonautonomous kernel functions.

These gaps motivate the development of new approaches that can discover both the analytical structure and parameters of distributed delay systems whilst preserving the sparsity and interpretability advantages of the SINDy framework.

4.1.3 Aim and objectives

The goal of this chapter is to develop a comprehensive methodology for sparse identification of DIDEs and REs that discovers the nonautonomous kernel function directly from time-series data without requiring prior knowledge of its structure. This chapter introduces a novel framework that combines integral SINDy with quadrature-based kernel approximation and external optimisation to enable systematic kernel discovery.

The main methodological innovations are as follows:

(*i*) **Quadrature-based kernel libraries**: a consistent approach for constructing candidate function libraries that explicitly include both the delay variable and state history, enabling recovery of nonautonomous kernels;

(*ii*) **Extended sparse regression**: an extension of the ISINDy framework that applies sparsity-promoting optimisation to identify the analytical structure of distributed memory kernels through weighted sums of quadrature terms;

(*iii*) **Joint optimisation of bounds and kernels**: external optimisation algorithms that simultaneously identify unknown integration bounds and other non-multiplicative parameters whilst maintaining interpretability through internal sparse regression;

(*iv*) **Comparative analysis of quadrature strategies**: investigation of various numerical quadrature rules and their effects on kernel recovery accuracy, computational cost, and robustness to noise.

The main objectives of this chapter are therefore as follows:

- to develop a general data-driven framework for identifying DIDEs and REs without requiring prior knowledge of kernel functional form or integration bounds;
- to extend the ISINDy framework to handle nonautonomous kernels via quadrature approximation and weighted library construction;
- to assess the performance of different quadrature rules, sample sizes, and sparsity parameters in kernel recovery;
- to show simultaneous identification of kernel structure and integration bounds via external and internal optimisation;
- to validate the proposed methodology through comprehensive numerical experiments on benchmark systems ranging from simple prototype models to complex biological systems.

4.1.4 Chapter organisation

The chapter is organised as follows. Section 4.2 introduces the mathematical foundations of DIDEs and REs, establishing notation and core concepts. Section 4.3 reviews the integral SINDy framework for ODEs, which provides the conceptual basis for the extension to distributed delays. Section 4.4 presents the main methodology for identifying distributed delay systems, detailing the construction of quadrature-weighted libraries and formulation of sparse regression, and offers a comparative analysis between the proposed distributed delay specialised approach and standard black-box SINDy application on a Ricker-type population model, highlighting the interpretability advantages of the new framework. Algorithmic implementations and external optimisation procedures are provided in this section. Section 4.5 presents comprehensive numerical experiments on prototype models, demonstrating the performance of the methodology under various conditions. Finally, Section 4.6 concludes the chapter, summarising contributions and limitations and outlining future research directions for data-driven discovery in infinite-dimensional dynamical systems.

4.2 Distributed delay differential and renewal equations

In many delayed dynamical systems, the time evolution of the concerned quantity depends not only on its values at just one or a few single distinct points in the past but rather on a weighted contribution of its history (full or possibly partial in between given or unknown time instants), with such weighting described by a delay kernel. Such systems are portrayed by models with distributed memory, in which past dependence is expressed via integral functionals. DIDEs and REs are primary classes of such models, together with relevant or even more complex formulations via PDEs (we refer the interested reader to [11, 52] as far as models of structured populations are concerned).

DIDE and RE

In principle, both DIDEs and REs can make use of a common structure for the right-hand side to capture the influence of the history of the system. In this respect a general DIDE is formulated as a differential equation

$$x'(t) = \Gamma(x_t) \tag{4.2.1}$$

whose rate of change depends on the past, whereas a typical RE

$$x(t) = \Gamma(x_t) \tag{4.2.2}$$

directly relates the current value of the unknown to its history x_t defined as $x_t(\sigma) := x(t+\sigma)$, $\sigma \in [-\tau, 0]$, for a given maximum delay $\tau > 0$. In both formulations, the central component is a functional $\Gamma : X \to \mathbb{R}^n$ for n a positive integer, which processes the state history and operates in a Banach space X of functions $[-\tau, 0] \to \mathbb{R}^n$ of suitable regularity (typically continuous for DIDEs and measurable for REs [46]). Γ accounts for the aggregated influence at the current time of the history of the system on the time span $[-\tau, 0]$[a]. This functional is, in general, defined by an integral

$$\Gamma(\varphi) := \int_{-\tau}^{0} \gamma(\sigma, \varphi(\sigma))\, \mathrm{d}\sigma, \quad \varphi \in X, \tag{4.2.3}$$

over the delay interval, where $\gamma : [-\tau, 0] \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth, possibly nonlinear map that we call the *kernel* function throughout the chapter. γ determines the specific nature of the model by assigning a weight to past states. The integration variable σ represents the history time from the present moment t: it directly affects the weight selection (first argument of γ) beyond the specific moment at which the past is considered (via the second argument of γ).

[a]We omit to consider formulations with multiple integrals because the following treatment can be generalised straightforwardly also thanks to the additivity of integration. Instead, in the case of DIDEs, we feel free to include next also current time and discrete delay terms.

Here $\Gamma : X \to \mathbb{R}^n$ denotes the distributed-delay functional and γ its kernel; they are unrelated to the gamma distribution mentioned later.

As we clarify in the following sections, this nonautonomous dependence of γ from σ

is a key aspect for its reconstruction from observed data. We instead conclude this section with a couple of prototype examples of DIDEs and REs.

Example 4.2: Logistic daphnia model

The *logistic daphnia model* [22, 26] given by

$$\begin{cases} b(t) = \beta S(t) \displaystyle\int_{a_*}^{a_\dagger} b(t-a)\, da, & (4.2.4a) \\ S'(t) = rS(t)\left(1 - \dfrac{S(t)}{\mathcal{K}}\right) - \nu S(t) \displaystyle\int_{a_*}^{a_\dagger} b(t-a)\, da, & (4.2.4b) \end{cases}$$

offers a simplified framework of an RE coupled to a DIDE to describe a consumer-resource dynamics. Above b denotes the birth rate of a population feeding on a resource S that in the absence of the consumer grows logistically with the growth rate r and carrying capacity $\mathcal{K}$. β and ν are the effective fertility and consumption rates of the adult individuals of the consumer population, which are assumed to have a constant survival probability between the maturation age a_* and the maximal age $a_\dagger$.

In Section 4.5 we employ (4.2.4) as a representative example of models of coupled DIDEs/REs, making it an ideal starting point for demonstrating the effectiveness of the proposed sparse identification of distributed memory kernels. In the experiments, we will treat both the maturation and the maximum ages as unknown. In terms of (4.2.1) and (4.2.2) (and recall footnote [a]) the right-hand side of (4.2.4) can be rewritten as

$$f(b_t, S_t) + \Gamma(b_t, S_t)$$

with

$$f(\varphi, \psi) := \left(0, r\psi(0)\left(1 - \frac{\psi(0)}{\mathcal{K}}\right)\right)^T$$

and

$$\Gamma(\varphi, \psi) := \int_{-a_\dagger}^{-a_*} \gamma(\sigma, \varphi(\sigma), \psi(\sigma))\, \mathrm{d}\sigma$$

for

$$\gamma(\sigma, \varphi(\sigma), \psi(\sigma)) := (\beta\psi(0)\varphi(\sigma), -\gamma\psi(0)\varphi(\sigma))^T.$$

In Section 4.5 we will identify both f and γ. Note that in this case, the kernel γ is *autonomous*.

Example 4.2: Behavioural epidemiology SIR model

In the *behavioural epidemiology framework* considered, e.g., in [7], the classical SIR paradigm can be expanded to include vaccination conditioned on information decision processes. Consider a stable and equally interacting population structured into compartments comprising susceptible (S), infected (I), recovered (R) and vaccinated (V) individuals, with the vaccination rate governed by an information variable $M(t)$ tracking past disease levels. A possible mathematical model reads

$$\begin{cases} S'(t) = \mu\,[1 - p(M(t))] - \mu S(t) - \beta S(t)I(t), \\ I'(t) = \beta S(t)I(t) - (\mu + \nu)I(t), \\ R'(t) = \nu I(t) - \mu R(t), \\ V'(t) = \mu p(M(t)) - \mu V(t), \end{cases}$$

where μ is the demographic turnover rate, β is the contact rate, and ν is the recovery rate. The coverage function $p(M)$ is assumed to be non-decreasing and is used to express the proportion of immunised newborns in terms of an information variable

$$M(t) := \int_{-\infty}^{0} K(\sigma)h(S(t+\sigma), I(t+\sigma))\,\mathrm{d}\sigma.$$

Here, $h : [0, +\infty) \times [0, +\infty) \to \mathbb{R}$ accounts for the prevalence of the disease weighted through a normalised memory kernel K. Typically K has the form of an *Erlang distribution*, but realistic data may call for different choices, asking for suitable identification tools.

Note that in general the local stability at the endemic equilibrium is highly affected by the properties of K [3]: as the memory concentration increases, the system is likely to have an increased number of intervals with instability in the form of Hopf bifurcations and oscillations. The model above considers an infinite delay and although analytical tools can handle this case in particular situations (e.g., resorting to the linear chain trick [93] as in [48]) as well as suitable quadrature rules on the half-line can be employed, time truncation can prove successful for the sake of identification, as we will show in Section 4.5 for another *(Ricker-type)* population model with infinite delay that we will consider first in Section 4.4.3.

4.3 Integral SINDy for ODEs

The SINDy framework of [30, 31] for ODEs seeks to recover parsimonious governing equations for dynamical systems directly from time-series data. When time derivatives of the state variables are not available, the standard approach [30, 31] relies on numerical differentiation, which may lead to significant sensitivity to the measurement noise. The alternative application of SINDy to the integral form of the underlying ODE has been proposed to improve robustness [96, 117, 140]. In this version, hereafter referred to as integral SINDy, the derivative estimation is replaced with numerical integration, which is an inherently more stable operation in noisy settings. This choice has inspired our treatment of systems with distributed memory, therefore we briefly describe its main features next.

Consider the IVP

$$\begin{cases} x'(t) = f(x(t)), & t \in [0,T], \\ x(0) = x_0, \end{cases} \tag{4.3.1}$$

for a general autonomous system of ODEs where $f : \mathbb{R}^n \to \mathbb{R}^n$ is an unknown vector field to be identified, $x_0 \in \mathbb{R}^n$ and $T > 0$. Integrating (4.3.1) yields the equivalent integral form

$$x(t) - x_0 = \int_0^t f\left(x(\sigma)\right) \mathrm{d}\sigma, \tag{4.3.2}$$

In practical scenarios, the state is observed at discrete times $t_1, \ldots, t_m$ and the corresponding snapshots are organised in the data matrix

$$\mathbf{X} = (X_1, \ldots, X_n) := \begin{pmatrix} x_1(t_1) & \cdots & x_n(t_1) \\ \vdots & \ddots & \vdots \\ x_1(t_m) & \cdots & x_n(t_m) \end{pmatrix} \in \mathbb{R}^{m \times n}. \tag{4.3.3}$$

Following the SINDy methodology, each component f_j of the vector field is approximated as a sparse linear combination of candidate functions from a predefined library $\Theta(x) \in \mathbb{R}^{m \times p}$. The library may include polynomial, trigonometric, or other nonlinear terms. Hence

$$f_j(x) \approx \Theta(x^T)\xi_j, \tag{4.3.4}$$

where $\xi_j \in \mathbb{R}^p$ is a sparse coefficient vector for the j-th governing equation. Substituting the latter into (4.3.2) requires integrating the candidate functions of the library. To this aim we resort to a simple rectangles formula based on the same time instants of the data sampling, assumed to be uniformly distributed in $[0, T]$ for the sake of simplicity. By letting $h := t_k - t_{k-1}$ we get

$$x_j(t_i) - x_{0,j} \approx \left(\int_0^{t_i} \Theta(x^T(\sigma)) \, \mathrm{d}\sigma \right) \xi_j \approx \left(h \sum_{k=0}^{i-1} \Theta(x^T(\sigma_k)) \right) \xi_j \tag{4.3.5}$$

for each state component $j = 1, \ldots, n$ and every time instant t_i, $i = 1, \ldots, m$. This establishes a relationship between the change in the state variable over the interval $[0, T]$ and a linear combination of the functions of Θ through the unknown coefficients ξ_j. According to the standard SINDy procedure, the regularised least squares regression

$$\hat{\xi}_j = \arg\min_{\xi \in \mathbb{R}^p} \left(\left\| \mathbf{X}_j - x_{0,j} - \left(h \sum_{k=0}^{i-1} \Theta(\mathbf{X}_{k,:}) \right) \xi \right\|_2 + \lambda \|\xi\|_1 \right) \tag{4.3.6}$$

is solved independently for each $j = 1, \ldots, n$ to find the sparse vector ξ_j that best describes the dynamics f_j. Above $\mathbf{X}_{k,:}$ denotes the k-th row of $\mathbf{X}$ and λ is a regularisation parameter that controls sparsity. (4.3.6) is a standard convex optimisation problem that can be solved efficiently using established tools such as STLS [30] or LASSO [132]. In general one can resort to other quadrature formulas and the approach can be easily adapted to the case of non-equispaced samples by using (piecewise) linear interpolation. Next we adopt the underlying use of quadrature to tackle equations with distributed delays, where in general a remarkable difference from what described here is that the integral kernel is by nature nonautonomous.

4.4 Extending SINDy to identify distributed memory kernels

We first illustrate in Section 4.4.1 our general approach to identify the kernel g of a distributed memory dependence that characterises the right-hand side (4.2.3) of a DIDE (4.2.1) or an RE (4.2.2). Then in Section 4.4.2 we briefly discuss relevant algorithms including the use of advanced optimisation tools to recover possible unknown delays or non-multiplicative parameters[1] . In the hereafter, we use the term "external" optimisation to refer to this specific problem, which in this context is mainly devoted to recovering unknown integration extrema beyond model parameters characterising special nonlinearities that might be included in the SINDy library in a physics-informed fashion. Finally, in Section 4.4.3 we compare the new approach with a standard SINDy aimed at just reconstructing the unknown dynamics as a "black-box" model, which does not provide any information on the underlying integration kernel. This analysis is performed on a specific DIDE with a nonlinear and nonautonomous kernel of Ricker type.

4.4.1 Quadrature and weighted libraries

In order to extend SINDy to recover distributed memory right-hand sides, it is key to first apply a suitable quadrature formula to approximate the integral representing $\Gamma(x_t)$ in (4.2.3) as a sum of weighted kernel values:

$$\int_{-\tau}^{0} \gamma(\sigma, x(t+\sigma))\,\mathrm{d}\sigma \approx \sum_{k=1}^{K} w_k \gamma(\sigma_k, x(t+\sigma_k)). \tag{4.4.1}$$

The above K represents the number of quadrature nodes $\sigma_k \in [-\tau, 0]$ and the corresponding weights w_k, $k = 1, \ldots, K$. According to the driving principle behind SINDy, we assume that the kernel γ admits a sparse linear representation within a library of candidate functions Θ. Since at each time t the kernel γ is a function of both the delay variable σ and the state value $x(t+\sigma)$, the library must take into account both arguments, thus allowing the recovery of nonautonomous kernels. Therefore, the delay variable σ is treated as an explicit candidate function, together with its possible nonlinear dependencies. Then, in view of the quadrature (4.4.1), for each $k = 1, \ldots, K$ we first consider a library Θ_k of candidate functions p which depend on the quadrature node σ_k and time-shifted state values $x(t+\sigma_k)$[2]. Following the usual SINDy description relying on the available time samples organised in the matrix $\mathbf{X}$ as in (4.3.3), such a k-th library may typically include polynomial terms up to a given degree d as

$$\begin{cases} \Theta_k := \Theta(\sigma_k, \mathbf{X}(\cdot + \sigma_k)) = \big[\, 1, \\ \qquad \sigma_k, \mathbf{X}_1(\cdot+\sigma_k), \ldots, \mathbf{X}_n(\cdot+\sigma_k), \\ \qquad \sigma_k^2, \sigma_k\mathbf{X}_1(\cdot+\sigma_k), \ldots, \sigma_k\mathbf{X}_n(\cdot+\sigma_k)), \mathbf{X}_1^2(\cdot+\sigma_k), \\ \qquad \ldots \\ \qquad \sigma_k^d, \sigma_k^{d-1}\mathbf{X}_1(\cdot+\sigma_k), \ldots, \mathbf{X}_n^d(\cdot+\sigma_k)\big], \end{cases}$$

plus any other nonlinear function of either σ_k or $\mathbf{X}_j(\cdot+\sigma_k)$, $j = 1, \ldots, n$, that the user can add based on prior model knowledge or any other available information (e.g., e^{σ_k},

[1]Multiplicative ones are indeed contained in the regression vectors ξ_j.

[2]Note that in principle p can depend on k even though we do not resort to such generality here.

$\sin(\mathbf{X}_j(\cdot + \sigma_k))$, etc.). Above $\mathbf{X}_j(\cdot + \sigma_k)$ represents the j-th state variable evaluated at shifted time instants $t_i + \sigma_k$, $i = 1, \ldots, m$: such values can be recovered by (piecewise linear) interpolation if not already at disposal within $\mathbf{X}$.

A weighted sum of these libraries Θ_k, $k = 1, \ldots, K$, through the chosen quadrature weights w_k represents the complete SINDy library to be used in the final regression problem. For an RE (4.2.2) the corresponding linear representation reads

$$\mathbf{X}_j \approx \left(\sum_{k=1}^{K} w_k \Theta(\sigma_k, \mathbf{X}(\cdot + \sigma_k)) \right) \xi_j, \tag{4.4.2}$$

for each j-th component of the state, $j = 1, \ldots, n$. For a DIDE (4.2.1) it is enough to replace $\mathbf{X}_j$ at the left-hand side with samples data $\mathbf{X}'_j$ for the time derivatives[3]. In (4.4.2) the vector $\xi_j \in \mathbb{R}^p$ is the sparse vector of coefficients sought that finally selects the active library terms for the j-th equation. To get each ξ_j we independently solve the regularised regression problem

$$\hat{\xi}_j = \arg\min_{\xi \in \mathbb{R}^p} \left(\left\| \mathbf{X}_j - \left(\sum_{k=1}^{K} w_k \Theta(\sigma_k, \mathbf{X}(\cdot + \sigma_k)) \right) \xi_j \right\|_2 + \lambda \|\xi_j\|_1 \right).$$

for each $j = 1, \ldots, n$. This is a standard convex optimisation problem that can be solved efficiently using established tools such as STLS [30] and LASSO [132]. By collecting all the resulting sparse vectors in a matrix $\Xi := (\hat{\xi}_1, \ldots, \hat{\xi}_n) \in \mathbb{R}^{p \times n}$, we finally identify (4.2.2) with

$$x(t) \approx \left(\int_{-\tau}^{0} \Theta(\sigma, x_t^T) \, \mathrm{d}\sigma \right) \Xi$$

(or (4.2.1) with $x'(t)$ replacing $x(t)$ at the left-hand side), thus revealing the underlying analytical structure of the kernel γ in (4.2.3).

4.4.2 Algorithmic implementation

We give here a brief description of the algorithms included in Appendix A that are used in Section 4.5 for numerical tests.

Algorithm A.8 outlines a systematic method for identifying DIDEs (4.2.1) and REs (4.2.2) using the proposed SINDy framework as illustrated in the previous section. The resulting tool relies on a priori knowledge of the involved delays, which provides the concerned integration window(s). In realistic situations, this information is hardly available, and the same consideration holds for any other (non-multiplicative) parameter characterising possible specific nonlinear dependencies (e.g., the exponent of an exponential function or the frequency or phase of a sinusoid), as well as for numerical parameters such as the number K or the distribution of the quadrature nodes. For this reason, following [16], we integrate the new SINDy framework of Algorithm A.8 into a second algorithm, viz. Algorithm A.9, which performs an iterative external optimisation loop over the parameters of interest to minimise the SINDy reconstruction error given by Algorithm A.8. We introduce the latter for REs as

$$\varepsilon(\rho) := \|\mathbf{X} - \Theta(\rho)\Xi(\rho)\|_2^2, \tag{4.4.3}$$

[3] Again, derivative values can be recovered from $\mathbf{X}$ via suitable techniques for noise reduction [37] if not available from measurements. Alternatively, one can resort to the integral reformulation of the DIDE as explained in Section 4.3 for ODEs.

where ρ denotes the vector of possible parameter values in the current step of Algorithm A.9 (for DIDEs, replace again $\mathbf{X}$ with $\mathbf{X}'$)[4].

External optimisation methods are used to effectively search the parameter space to reduce the error function to a value such that both the recovered kernel and its associated parameters are properly aligned with the observed data. Therefore, Algorithm A.9 helps uncover not only the sparse structure of the distributed delay kernel (via the "internal" regression of SINDy) but also its bounds, other embedded parameters, and possible numerical settings (via the "external" optimisation loop), thereby enhancing both the interpretability and the predictive accuracy of the identified model.

4.4.3 Comparing kernel reconstruction to a black-box SINDy approach

To illustrate the differences between applying the proposed specialised framework for distributed delays (hereinafter named DD-SINDy) and a standard SINDy "black-box" approach (hereinafter named BB-SINDy)

Example 4.4.3: Ricker-type population model

We consider Ricker-type population model which represents the DIDE [9, 75]. The model reads

$$x'(t) = d_0 \left(\int_{-\infty}^{0} F_{n/\tau}^{(n)}(\sigma) e^{\eta + d_1\sigma - a x(t+\sigma)} x(t+\sigma)\, \mathrm{d}\sigma - x(t) \right), \tag{4.4.4}$$

modelling the growth of a single-species population according to a gamma-distributed maturation delay and a Ricker-type nonlinear reproduction with positive parameters η and a (see [9, equation (4)] or [75, 111]). $x(t)$ denotes the size of adult individuals at time $t \geq 0$, $d_0 > 0$ the constant death rate of the population, and $d_1 \geq 0$ the constant death rate during the maturation stage.

The integral term represents the birth rate, which depends on the history of the population, with a delay continuously distributed according to

$$F_{\alpha}^{(n)}(\sigma) := \frac{\alpha^n (-\sigma)^{n-1} e^{\alpha\sigma}}{(n-1)!}, \quad \sigma \in (-\infty, 0], \tag{4.4.5}$$

for n a positive integer and $\alpha := n/\tau$, with mean $\tau > 0$ and variance τ^2/n. Our objective is to recover the nonautonomous kernel

$$\gamma(\sigma, \varphi(\sigma)) := d_0 F_{\alpha}^{(n)}(\sigma) e^{\eta + d_1\sigma - a\varphi(\sigma)} \varphi(\sigma), \quad \sigma \in (-\infty, 0], \tag{4.4.6}$$

using the time-series data of $x(t)$. For the sake of focusing on the comparison of the DD-SINDy with BB-SINDy, in this section we deliberately tackle a simple instance of (4.4.4) by choosing $d_1 = 0$, $d_0 = a = 1$, $\eta = \log 80$, $n = 4$ and $\tau = 1$, in which case $\alpha = 4$ and the explicit kernel (4.4.6) reduces to

$$\gamma(\sigma, \varphi(\sigma)) := \zeta \cdot \sigma^3 e^{4\sigma} e^{-\varphi(\sigma)} \varphi(\sigma) \tag{4.4.7}$$

[4] Note that in the experiments, part of the data are used to train SINDy and part to validate the recovered model; in this respect (4.4.3) is extended to account for both contributions.

for $\zeta := -10\,240/3$. The above choices return $\gamma(\sigma, x(t+\sigma))$ as a product of integer powers of the basis functions σ, e^{σ}, $e^{-x(t+\sigma)}$, $x(t)$ and $x(t+\sigma)$, which we therefore use as candidates to construct a polynomial SINDy library Θ_{DD} of degree (at least) 4 for DD-SINDy. Moreover, we further simplify model (4.4.4) by truncating the influence of the history on the integration window $[-10\tau, 0]$ (in Section 4.5.2 we consider a more general case with non-integer powers as well as irrational exponents to challenge the external optimisation Algorithm A.9 in combination with kernel recovery via DD-SINDy).

As far as BB-SINDy is concerned, (4.4.4) is seen just as an instance of the general form (4.2.1), so that we seek for a sparse representation of the right-hand side as

$$\Gamma(x_t) \approx \Theta_{\mathrm{BB}}(x_t^T)\Xi.$$

By resorting to a general polynomial library it is clear that one expects BB-SINDy to correctly recover the coefficient $-d_0$ of the linear term in (4.4.4), but in general all the other coefficients will not serve to identify interpretable structures for the right-hand side, although the returned model can well serve for make predictions via simulation beyond the time range of available data.

For the sake of comparison, synthetic time series data for the population x were generated in the time window $[0, 20]$ by numerically solving the IVP for (4.4.4) with initial function $\varphi(s) = 0.5$, $s \in [-10\tau, 0]$. Actually MATLAB's `dde23` was used to integrate a version of (4.4.4) with the integral replaced by a uniform rectangles-based quadrature with $K = 50$ uniform nodes. Then 100 samples were selected, with 80% reserved for training both DD-SINDy and BB-SINDy, while the remaining data were used for validation to assess the accuracy of the prediction. The same number and distribution of quadrature nodes was used for DD-SINDy, while the sparsity-promoting hyperparameter was set to $\lambda = 10^{-2}$ for both DD-Sindy and BB-SINDy. STLS was used to solve the relevant sparse regression problem. The trajectories of the two recovered models are compared to the ground truth in Figure 4.4.1, showing a remarkable reconstruction capacity for both training and validation.

The root mean square error (RMSE) (4.4.3) measured on the derivatives was also used to evaluate the performance of both approaches, see Table 4.4.1. The latter shows also the accuracy of the coefficients learnt during sparse regression. DD-SINDy is able to recover both d in (4.4.4) and γ in (4.4.7), while BB-SINDy returns only the former as already explained.

4.5 Numerical experiments

In this section we present the results of quantitatively assessing the proposed methodology on three different models by using Algorithm A.8 and Algorithm A.9 discussed in Section 4.4.2 and implemented in Appendix A.

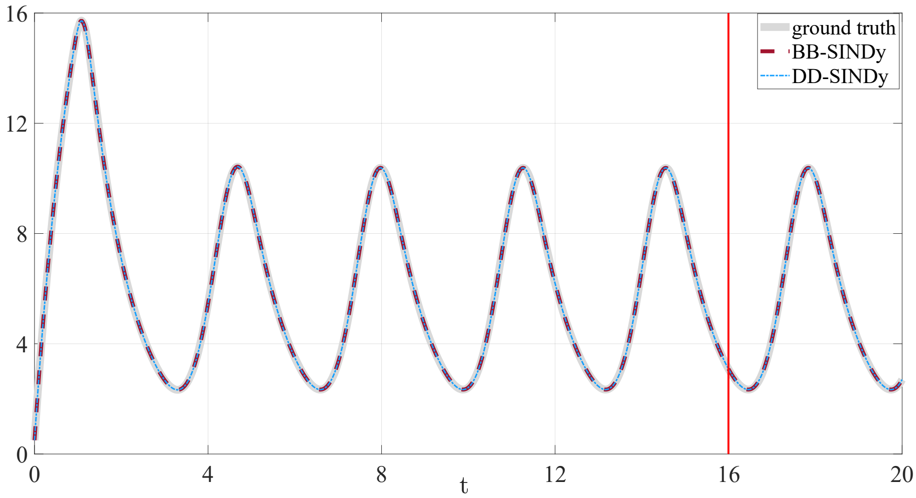

Figure 4.4.1: comparison of trajectories of (4.4.4) reconstructed by DD-SINDy and BB-SINDy. $m = 100$ uniform samples were collected in the time window $[0, 20]$ with $K = 50$ and $\lambda = 10^{-2}$, using 80% for training and 20% for validation.

SINDy	$\lvert d_0 - \widehat{d_0}\rvert$	$\lvert\gamma - \widehat{\gamma}\rvert$
BB-SINDy	6.81×10^{-2}	-
DD-SINDy	3.29×10^{-3}	4.92×10^{-3}
performance metrics		
$\mathrm{RMSE}_{x'}$	training	validation
BB-SINDy	3.51×10^{-2}	9.96×10^{-2}
DD-SINDy	4.36×10^{-2}	3.46×10^{-2}

Table 4.4.1: comparison of DD-SINDy with BB-SINDy in terms of the absolute error on the reconstructed coefficients (top) and RMSE (4.4.3) on derivatives (bottom) for (4.4.4) in the time window $[0, 20]$ using $m = 100$ samples and $\lambda = 10^{-2}$, with DD-SINDy using $K = 50$ quadrature nodes.

4.5.1 A logistic renewal equation

Example 4.5.1: Toy RE model

We assess the proposed methodology using prototype equations such as

$$x(t) = \int_{-3}^{-1} (\sigma + 1)x(t + \sigma)[1 - x(t + \sigma)]\, d\sigma. \tag{4.5.1}$$

Data is generated using the time integration scheme of [97] with a constant history function $\varphi(\sigma) = 0.5, \sigma \in [-3, 0]$. Uniformly spaced samples are collected on $[0, 20]$, using the first 50% for training and the remainder for validation.

A polynomial SINDy library of degree 3 is constructed in combination with trapezoidal quadrature [138] and the tests are carried out by varying the number K of quadrature nodes, the hyperparameter promoting sparsity λ and the number m of samples.

Table 4.5.1 presents the active library terms recovered by DD-SINDy with $m = 100$, $K = 128$ and $\lambda = 10^{-5}$, together with relevant RMSE on the derivative for both training and validation. The quadratic library proves insufficient to capture the system dynamics, while extending the library to $d = 4$ slightly degrades parameter accuracy due to the inclusion of redundant higher order terms. Conversely, the cubic library achieves the best performance, reducing the coefficient error to 10^{-3}. Correspondingly, Figure 4.5.1 illustrates the reconstructed trajectories.

library terms	$\lvert\xi_i - \widehat{\xi}_i\rvert$		
	Polynomial order 2	Polynomial order 3	Polynomial order 4
$x(t+\sigma)$	6.11×10^{-1}	1.82×10^{-3}	2.69×10^{-1}
$\sigma \cdot x(t+\sigma)$	6.90×10^{-1}	3.06×10^{-3}	4.62×10^{-1}
$x(t+\sigma)^2$	4.19×10^{-2}	2.74×10^{-3}	1.30×10^{-2}
σ^2	9.89×10^{-1}	-	-
$\sigma^2 \cdot x(t+\sigma)$	-	9.00×10^{-3}	2.55×10^{-1}
$\sigma \cdot x(t+\sigma)^2$	-	3.55×10^{-4}	9.93×10^{-3}
$x(t+\sigma)^3$	-	-	5.13×10^{-3}
$\sigma^3 \cdot x(t+\sigma)$	-	-	4.19×10^{-2}
$\sigma^2 \cdot x(t+\sigma)^2$	-	-	1.64×10^{-3}
$\sigma \cdot x(t+\sigma)^3$	-	-	1.90×10^{-3}
$x(t+\sigma)^4$	-	-	8.88×10^{-4}
performance metrics			
$\text{RMSE}_{x'}$			
training	2.01×10^{-1}	1.40×10^{-3}	1.25×10^{-3}
validation	1.44×10^{-1}	2.18×10^{-4}	3.70×10^{-4}

Table 4.5.1: absolute errors on the coefficients of the polynomial library terms recovered by DD-SINDy on (4.5.1) with degree $d = 2, 3$ and 4 (top); relevant RMSE on the derivative (bottom). $m = 100$ uniform samples were collected in the time window $[0, 20]$ with $K = 128$ and $\lambda = 10^{-5}$, using 50% for training and 50% for validation.

Further tests were carried out by varying the number K of quadrature nodes, the hyperparameter λ promoting sparsity and the number m of samples, employing also rectangle and Clenshaw–Curtis quadrature [41, 57, 134] for the sake of performance comparison, see Figures 4.5.2 to 4.5.5. In general and unsurprisingly, the accuracy improves by increasing the number K of quadrature nodes independently of the fixed value of λ and of the fixed number of samples m, see Figure 4.5.2 for $m = 50$, Figure 4.5.3 for $m = 200$ and Figure 4.5.4 for $m = 1\,000$. The behaviour of different quadrature formulas is similar for low or moderate values of m, while for larger values differences emerge, with Clenshaw-Curtis prevailing on the other choices. At fixed number of quadrature nodes K the accuracy decreases with the number of samples m, apparently reaching a barrier already with moderate values on the order of hundreds. Clenshaw-Curtis and trapezoidal rules perform a little better than rectangles, even though no clear trend is visible along

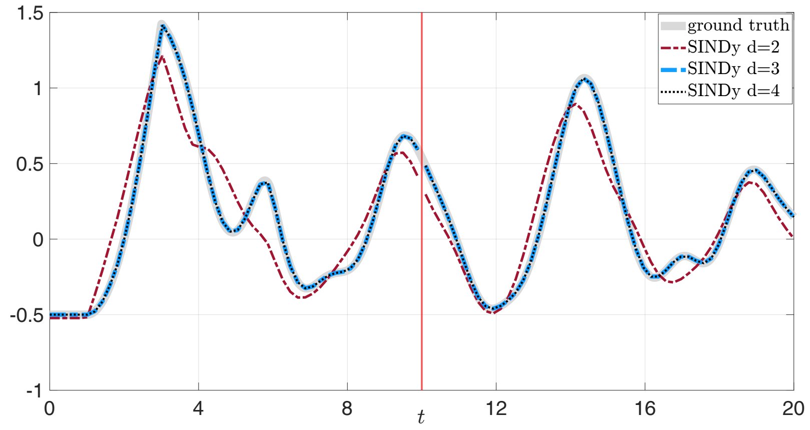

Figure 4.5.1: comparison of trajectories of (4.5.1) reconstructed by DD- SINDy with a polynomial library of degree $d = 2, 3$ and 4; corresponding data in Table 4.5.1.

the barrier.

The accurate identification of distributed delay kernels via DD-SINDy turns out to be a delicate balance among the adopted quadrature, an appropriate choice of λ and an adequate sampling (i.e., m). Let us just notice indeed that while the classical quadrature error is the result of replacing the integrand with a suitable finite-dimensional approximation, typically (piecewise) polynomial on given sets of nodes, in the case of sparse regression one has to take into account also (and mainly) for the linear approximation through the library candidate functions.

4.5.2 An advanced Ricker-type model

As in our second case study, our method is applied to a more sophisticated system as an advanced model (4.4.4). With parameter settings $n = 4, a = \pi/10, d_1 = 0.5$ and $\tau = 1$, so that the kernel (4.4.6) becomes

$$\gamma(\sigma, \varphi(\sigma)) := \zeta \cdot \sigma^3 e^{4.5\sigma} e^{-\pi\varphi(\sigma)/10} \varphi(\sigma). \tag{4.5.2}$$

Therefore, the factor σ^3 can be recovered via a polynomial library that includes the candidate function σ, the two exponential factors need for external optimisation of both non-multiplicative parameters d_1 and a when these are assumed to be unknown as it is the case here. In particular Algorithm A.9 was used to externally optimise all the unknown values $\tau = 1$, $n = 4$, $\alpha + d_1 = 4.5$ and $a = \pi/10$. The recovered value for n was rounded to the nearest integer and combined with the optimised τ to get first $\alpha = n/\tau$ and then d_1 from the optimised value of $\alpha + d_1$, while a was optimised independently.

Data for the tests is generated similarly as illustrated in Section 4.4.3, collecting now $m = 500$ equidistant samples on $[0, 20]$, using the first 80% for training and the rest for validation. A polynomial SINDy library of degree $d = 4$ is employed in combination with trapezoidal quadrature with $K = 100$ nodes and $\lambda = 10^{-2}$. Table 4.5.2 collects the results concerning the externally optimised values (top part), the absolute error on the recovered sparse coefficients (central part) and the RMSE on the derivative both for training and validation (bottom part). Correspondingly, Figure 4.5.6 shows the recovered trajectory

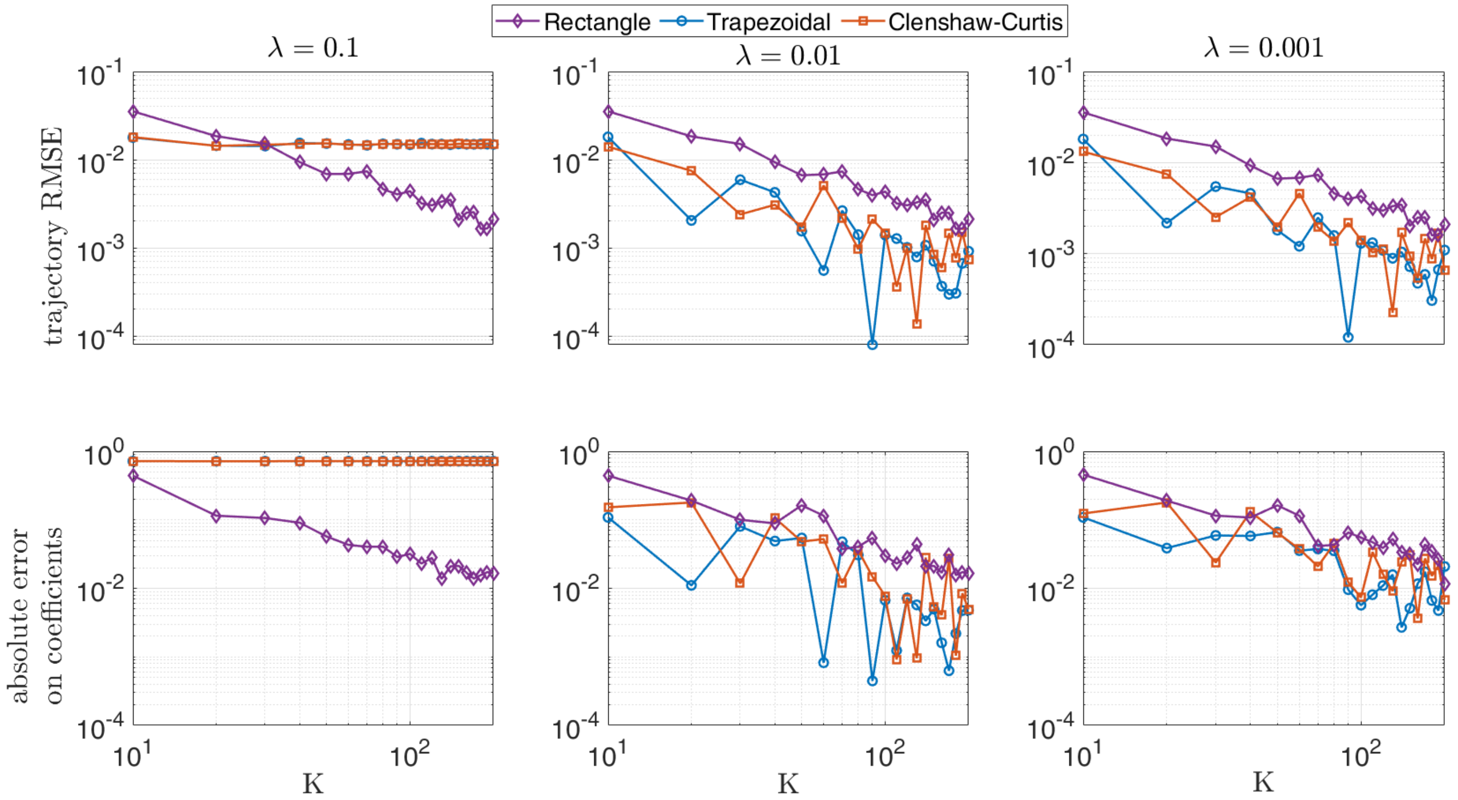

Figure 4.5.2: comparison of RMSE on trajectories (top row) and absolute error on coefficients (bottom row) for (4.5.1), varying K and $\lambda = 0.1$ (left), $\lambda = 0.01$ (center) and $\lambda = 0.001$ (right). $m = 50$ uniform samples were collected in the time window $[0, 20]$ using 50% for training and 50% for validation.

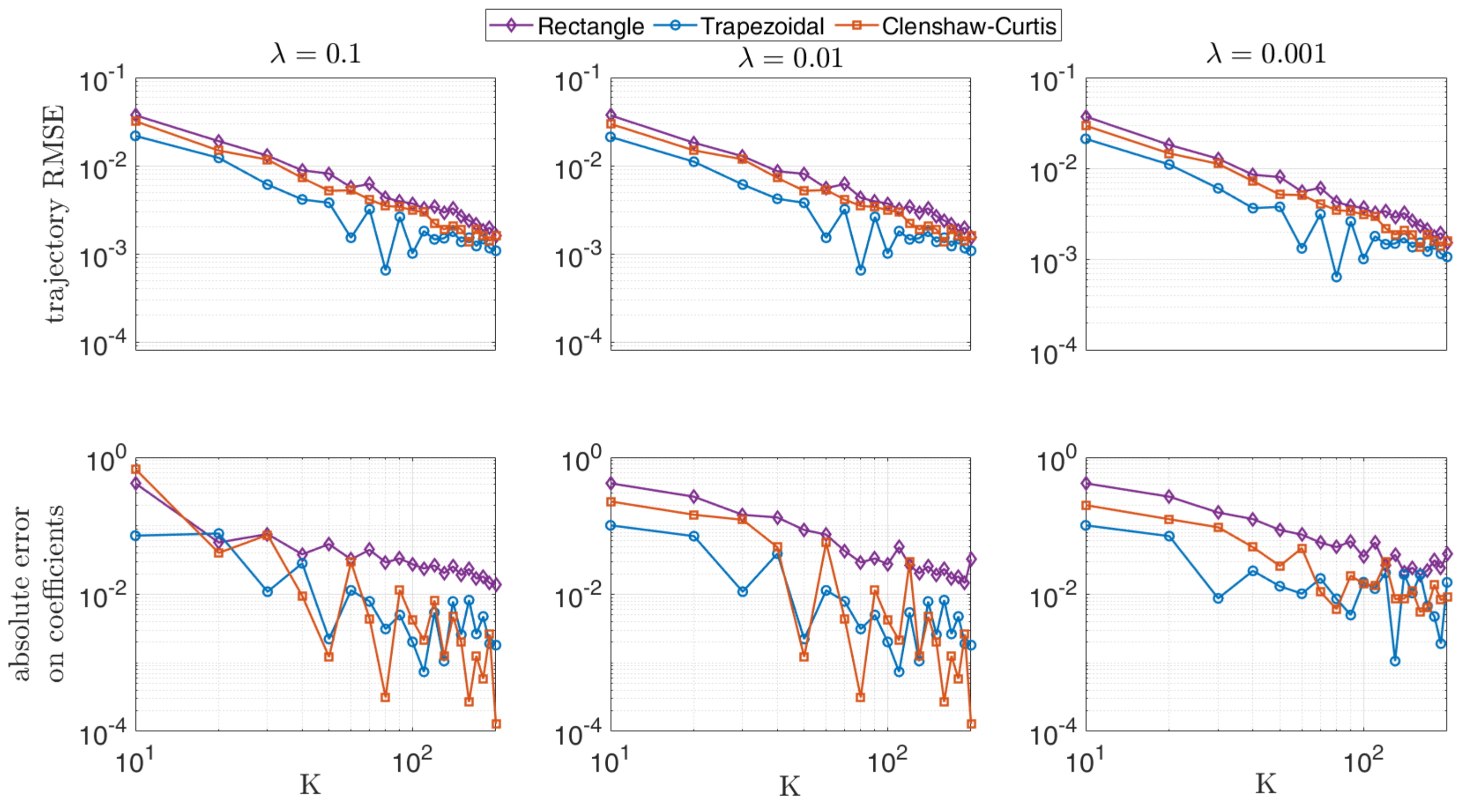

Figure 4.5.3: comparison of RMSE on trajectories (top row) and absolute error on coefficients (bottom row), varying K and $\lambda = 0.1$ (left), $\lambda = 0.01$ (center) and $\lambda = 0.001$ (right). $m = 200$ uniform samples were collected in the time window $[0, 20]$ using 50% for training and 50% for validation.

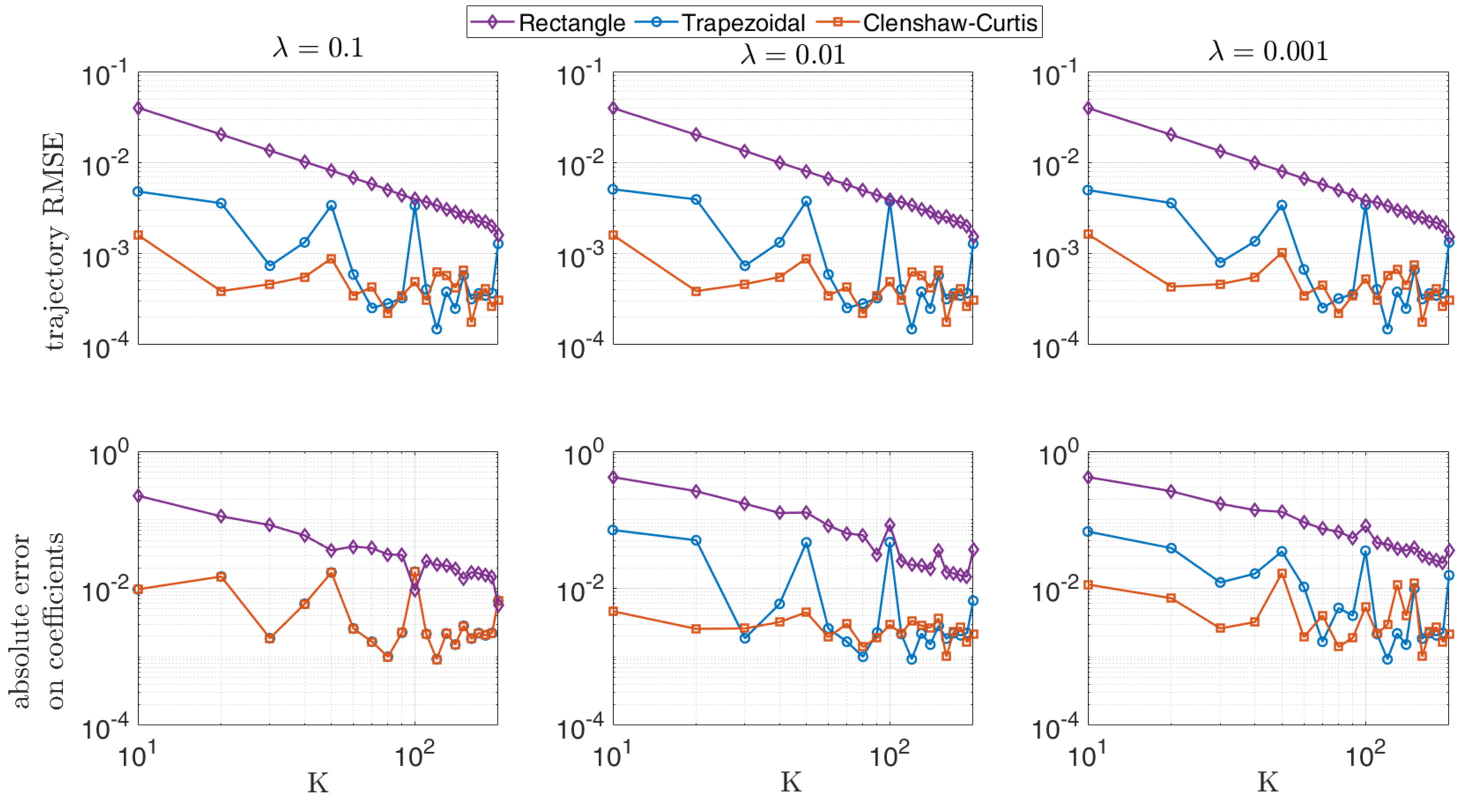

Figure 4.5.4: comparison of RMSE on trajectories (top row) and absolute error on coefficients (bottom row), varying K and $\lambda = 0.1$ (left), $\lambda = 0.01$ (center) and $\lambda = 0.001$ (right). $m = 1000$ uniform samples were collected in the time window $[0, 20]$ using 50% for training and 50% for validation.

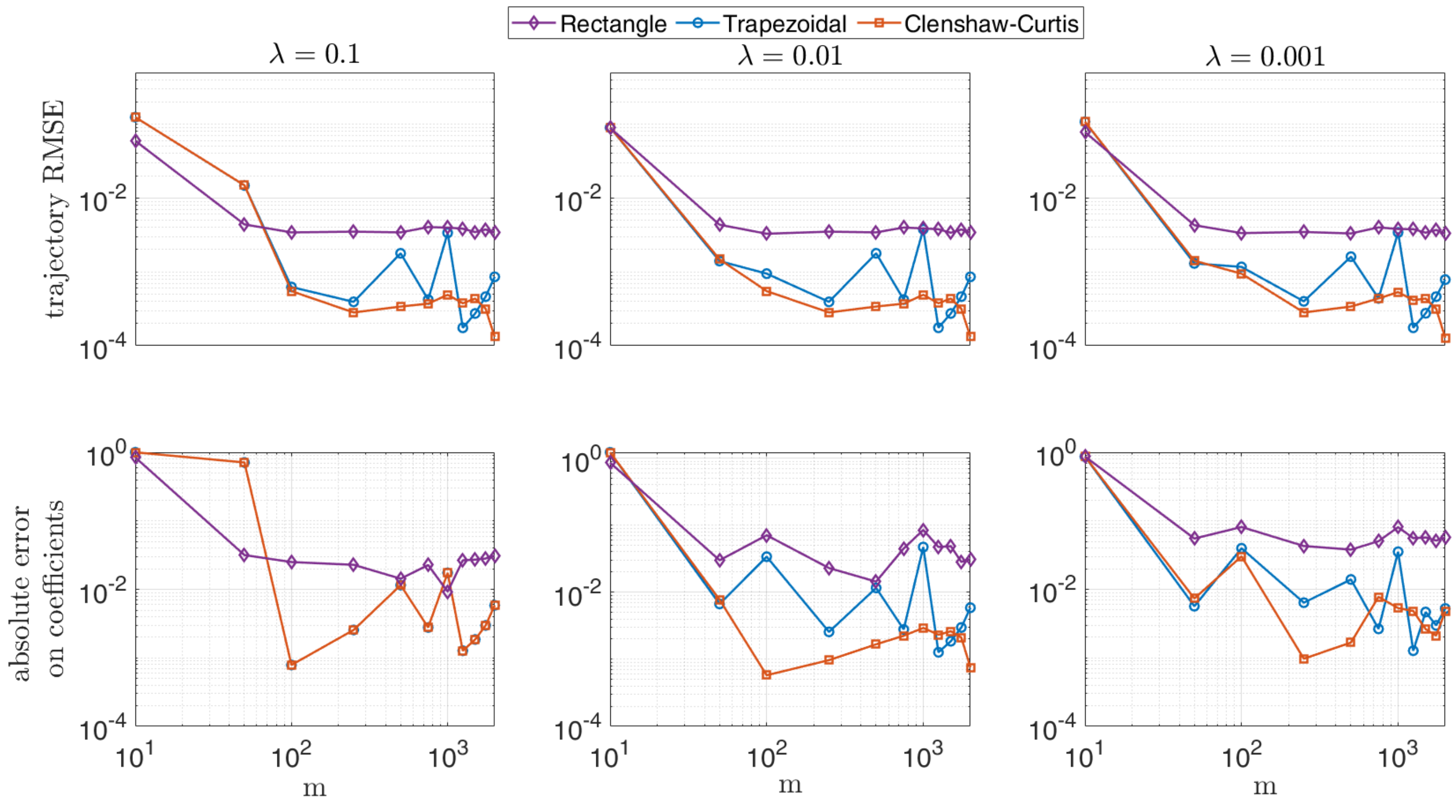

Figure 4.5.5: comparison of RMSE on trajectories (top row) and absolute error on coefficients (bottom row), varying m and $\lambda = 0.1$ (left), $\lambda = 0.01$ (center) and $\lambda = 0.001$ (right). $K = 100$ uniform samples were collected in the time window $[0, 20]$ using 50% for training and 50% for validation.

and kernel, while Figure 4.5.7 illustrates the iterations of the external optimisation by Algorithm A.9 as explained above.

The results in Table 4.5.2 show that the proposed methodology efficiently compresses significant model terms that contain small coefficients and estimation errors and demonstrates strong performance in generalisation and parameter estimation. Optimisation took approximately 1.50×10^3 s for 971 calls to the DD-SINDy to reach a given tolerance of 10^{-4}, ensuring that PS efficiently identified high-dimensional dynamical structures.

As illustrated in Figure 4.5.6, the recovered trajectories closely resemble the ground truth, and the extracted kernel agrees strongly with the actual kernel, thus confirming the approach for the (4.4.4) data. In addition, Figure 4.5.7 illustrates the convergence behaviour of the extracted parameters $(\widehat{n}, \widehat{\tau}, \widehat{d_1}, \widehat{a})$, where the optimisation paths demonstrate effective convergence and alignment with the true values. The results obtained here are related to a noise-free dataset.

Similar trends are observed for Figures 4.5.8 and 4.5.9 in the noisy dataset with 20% noise, for which the method showed strong generalisability and effective parameter retrieval, despite added perturbations. The results based on Table 4.5.3 demonstrate that the proposed technique greatly summarises important model terms containing negligible coefficients and minimal estimation errors, thus retaining high generalisation and parameter estimation performance despite noisy conditions. Optimisation took approximately 2.04×10^3 s for 874 solver calls.

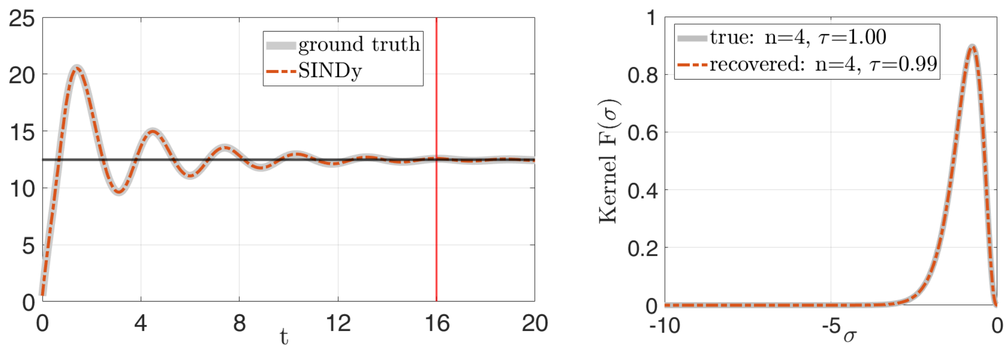

Figure 4.5.6: comparison of trajectories (left) and kernel functions over the delay interval $[-10\tau, 0]$ (right) of (4.4.4) with (4.5.2) by DD-SINDy; corresponding data in Table 4.5.2, horizontal line in the left panel indicates the equilibrium.

4.5.3 A logistic daphnia model

As a third case study, we applied our methodology to a more complex system, a simplified logistic Daphnia model (4.2.4). For a system with state variables b and S and parameters $a_*, a_\dagger, \beta, r, \mathcal{K}$ and ν, a transcritical bifurcation occurs at the consumer free equilibrium $(0, \mathcal{K})$. With the parameter values $a_* = 3, a_\dagger = 4, r = 1, \mathcal{K} = 1$ and $\nu = 1$, this bifurcation occurs at the value of the critical parameter $\beta = (\mathcal{K}(a_\dagger - a_*))^{-1} = 1$. At this point, a new, positive, and stable equilibrium has emerged. This new equilibrium subsequently undergoes a Hopf bifurcation at a higher value of $\beta \approx 3.0161$.

To identify the governing equations from the data, we constructed a SINDy candidate library comprising polynomial terms up to degree two in $b(t)$ and $S(t)$ in combination

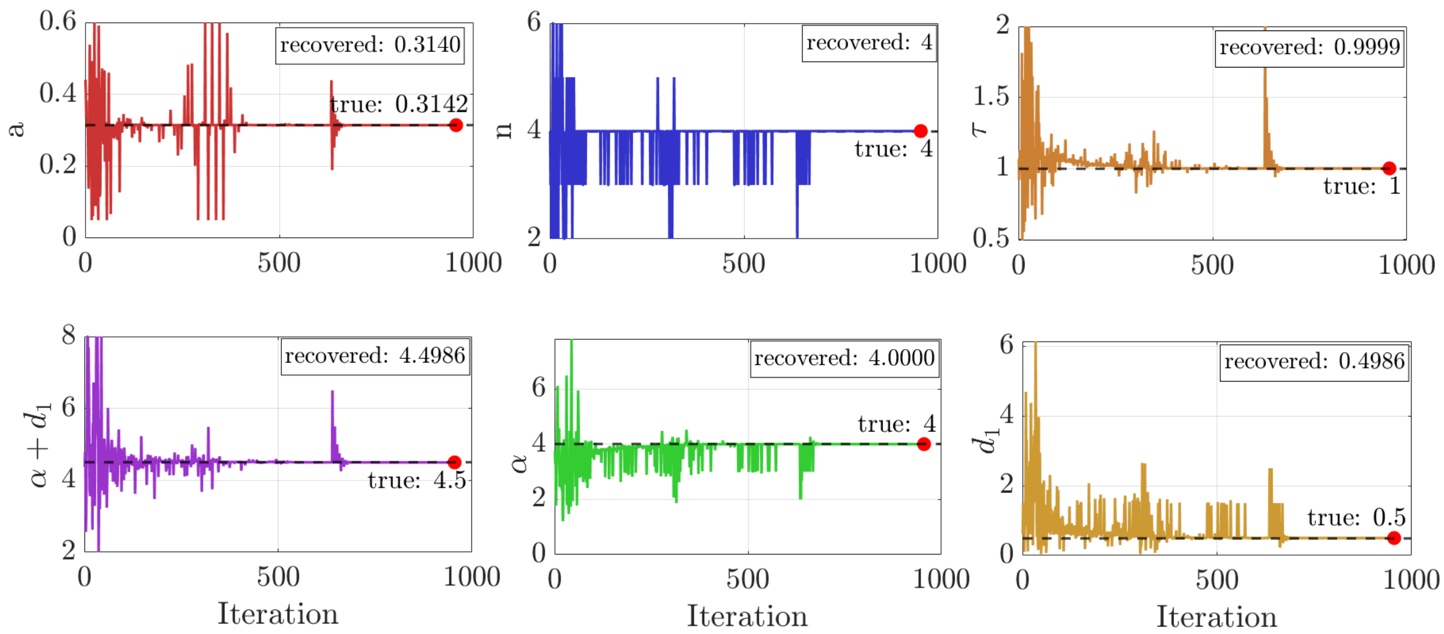

Figure 4.5.7: plots of optimised unknown parameters $(\widehat{n}, \widehat{a}, \widehat{\tau}, \widehat{d}_1, \widehat{\alpha} + \widehat{d}_1, \widehat{\alpha})$ using particle swarm (PS) for the model (4.4.4). Red circle marks the true optimised values of the parameters.

error on externally optimised values	$\lvert n-\widehat{n}\rvert$ 0	$\lvert d_1-\widehat{d_1}\rvert$ 1.43×10^{-3}	$\lvert\tau-\widehat{\tau}\rvert$ 1.85×10^{-5}	$\lvert a-\widehat{a}\rvert$ 3.64×10^{-4}
error on library coefficients	$\lvert d_0-\widehat{d_0}\rvert$ 5.26×10^{-3}	$\lvert\zeta-\widehat{\zeta}\rvert$ 1.68×10^{-2}		
$\mathrm{RMSE}_{x'}$	training 1.15×10^{-3}	validation 2.74×10^{-4}		

Table 4.5.2: absolute errors on the optimised values for (4.4.4) (top) and on coefficients of the polynomial library terms recovered by DD-SINDy on (4.4.4)(middle); relevant RMSE on the derivative (bottom). $m = 500$ uniform samples were collected in the time window $[0, 20]$ with $K = 100$ and $\lambda = 10^{-2}$, using 80% for training and 20% for validation.

error on externally optimised values	$\lvert n-\widehat{n}\rvert$ 0	$\lvert d_1-\widehat{d_1}\rvert$ 1.48×10^{-3}	$\lvert\tau-\widehat{\tau}\rvert$ 1.10×10^{-4}	$\lvert a-\widehat{a}\rvert$ 3.83×10^{-4}
error on library coefficients	$\lvert d_0-\widehat{d_0}\rvert$ 4.02×10^{-3}	$\lvert\zeta-\widehat{\zeta}\rvert$ 6.75×10^{-2}		
$\mathrm{RMSE}_{x'}$	training 1.19×10^{-3}	validation 2.83×10^{-4}		

Table 4.5.3: absolute errors on the optimised values for noisy data (20%) of (4.4.4) (top) and on coefficients of the polynomial library terms recovered by DD-SINDy on (4.4.4)(middle); relevant RMSE on the derivative (bottom). $m = 500$ uniform samples were collected in the time window $[0, 20]$ with $K = 100$ and $\lambda = 10^{-2}$, using 80% for training and 20% for validation.

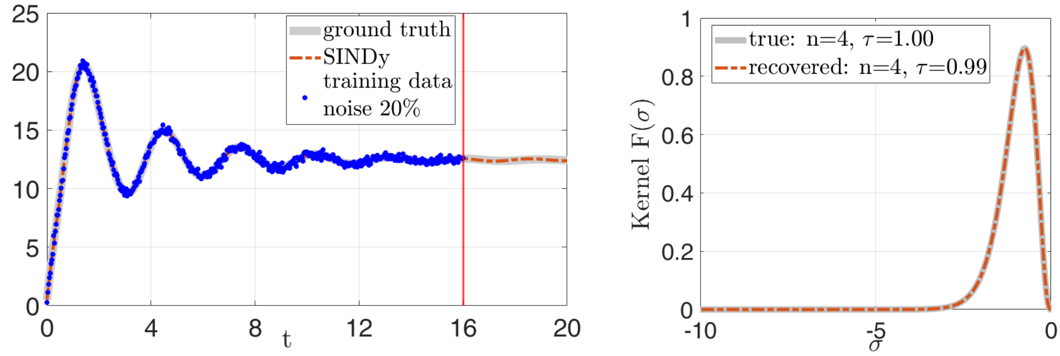

Figure 4.5.8: comparison of trajectories of (4.4.4) reconstructed by DD-SINDy with a polynomial library with 20% noise (left) and kernel functions over the delay interval σ (right); corresponding data in Table 4.5.3.

with trapezoidal quadrature. Time-series data are generated over the interval $t \in [0, 50]$, resulting in $m = 1733$ non-uniform samples using the MATCONT package [90]. This dataset is partitioned into an 80% training set and a 20% validation set. A notable aspect of our approach is that the integral bounds, a_* and $a_\dagger$, are not assumed to be known but are identified externally [16, 116] using the PS algorithm.

Using optimal bounds, the SINDy model is trained using a sparsity threshold of $\lambda = 10^{-3}$. Table 4.5.4 presents the active model terms that are successfully recovered using this method. Furthermore, Figure 4.5.10 illustrates the excellent agreement between the dynamics of the original system and the reconstructed trajectories of the discovered model, which validates the accuracy of our approach for the data from the logistic Daphnia model (4.2.4).

error on externally optimised values	$\lvert a_* - \widehat{a}_* \rvert$ 1.00×10^{-2}	$\lvert a_\dagger - \widehat{a}_\S \rvert$ 1.70×10^{-2}		
error on library coefficients	(4.2.4a) $S(t)b(t-a)$ 6.05×10^{-2}	(4.2.4b) $S(t)$ 3.01×10^{-3}	(4.2.4b) $S(t)^2$ 4.50×10^{-3}	(4.2.4b) $S(t)b(t-a)$ 4.44×10^{-2}
$\text{RMSE}_{x'}$	training 7.30×10^{-4}	validation 1.40×10^{-4}		

Table 4.5.4: absolute errors on the optimised values for (4.2.4) (top) and on coefficients of the polynomial library terms recovered by DD-SINDy on (4.4.4) with degree $d = 2$ (middle); relevant RMSE on the derivative (bottom). $m = 1733$ non-uniform samples were collected in the time window $[0, 50]$ with $K = 8$ and $\lambda = 10^{-3}$, using 80% for training and 20% for validation.

The findings from the logistic Daphnia model (4.2.4) underscore two key benefits of the proposed DD-SINDy framework. Firstly, the approach is effective in identifying coupled systems where the dynamics of one state variable (the resource S) influence the delayed feedback of another (the consumer birth rate b). Secondly, and perhaps more im-

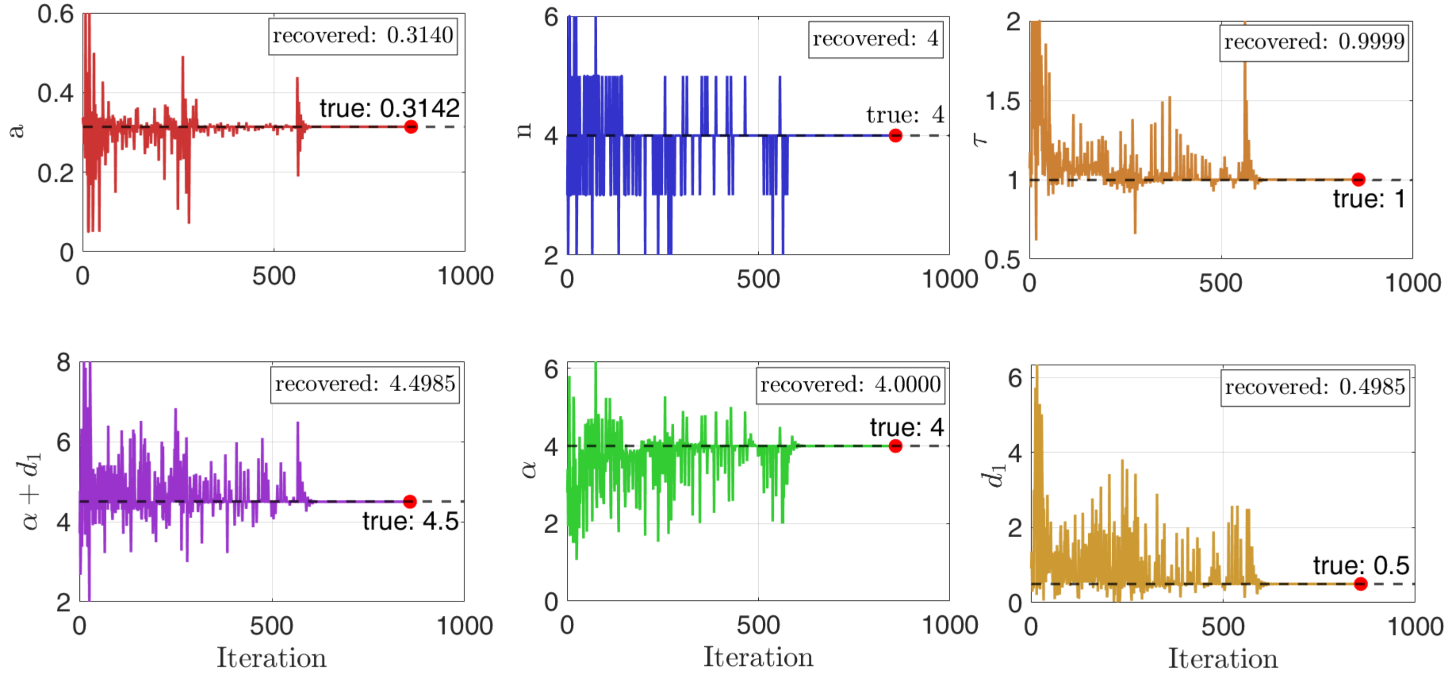

Figure 4.5.9: plots of the optimised unknown parameters $(\widehat{n}, \widehat{\tau}, \widehat{d}_1, \widehat{a})$ were generated using particle swarm optimisation (PS) for the noisy data of model (4.4.4). The red circle indicates the true optimised values of the parameters.

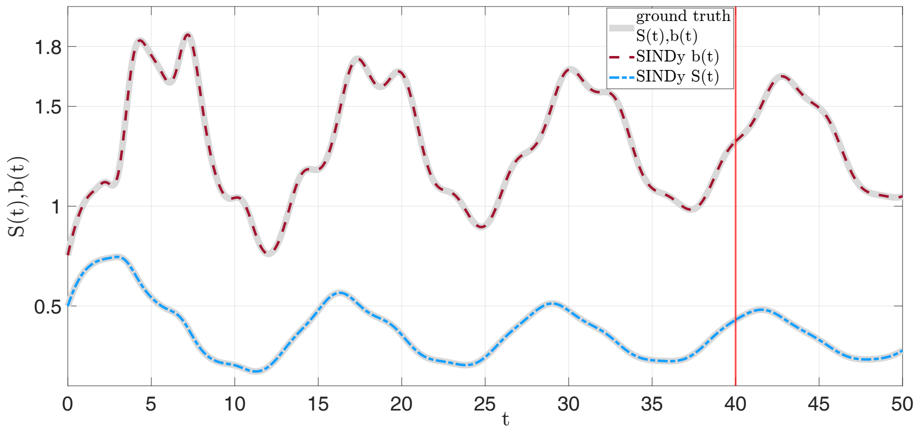

Figure 4.5.10: comparison of trajectories of (4.2.4) reconstructed by DD- SINDy with a polynomial library of degree $d = 2$; corresponding data in Table 4.5.4.

portantly, the algorithm adeptly determines the finite integration window $[a_*, a_\dagger]$. Unlike the Ricker model (4.4.4), which involves an infinite delay, this scenario necessitates pinpointing accurate boundaries of the memory kernel. The accurate reconstruction of the maturation age a_* and the maximum age $a_\dagger$ (refer to Table 4.5.4) illustrates that the external optimisation loop (Algorithm A.9) can accurately identify the 'window of influence' in historical data, offering not only a predictive model but also biologically meaningful insights into the characteristics of life-history of the population.

4.6 Conclusions

This chapter significantly extends the SINDy framework by expanding its applicability to identify the nonautonomous kernel functions of governing equations for systems with distributed delays and renewal equations. By combining a quadrature-based numerical approximation with sparse regression techniques, the proposed methodology directly reconstructs distributed delay kernels from time-series data. This development captures the complex effects of distributed memory inherent in infinite-dimensional dynamical systems.

The proposed methodology introduces several key innovations. First, we develop candidate function libraries that explicitly depend on both delay and state variables, enabling recovery of interpretable kernel structures instead of producing black-box high-dimensional models. Second, the framework applies optimisation to discover parsimonious representations of distributed memory kernels via weighted sums of quadrature terms. Third, external optimisation algorithms jointly identify unknown integration bounds alongside kernel structures, maintaining interpretability. Fourth, comparative analysis of various quadrature rules provides practical guidance for practitioners.

The novel nature of this work merits emphasis: this study represents the first general-purpose framework for identifying DIDEs and REs using data-driven sparse identification techniques. Whilst prior theoretical and numerical work has analysed distributed delay systems under the assumption of known functional forms, we present the first systematic approach capable of discovering both kernel analytical structure and integration bounds directly from time-series observations without requiring extensive prior knowledge.

The main contributions of this chapter are:

- development of a general data-driven framework for identifying DIDEs and REs without requiring prior knowledge of kernel functional form or integration bounds, representing a novel methodological advance in the treatment of distributed delay systems;
- extension of the ISINDy to handle nonautonomous kernels via quadrature approximation and weighted library construction, establishing a bridge between finite-dimensional sparsity methods and infinite-dimensional dynamical systems;
- comprehensive assessment of the performance of different quadrature rules, including rectangle, trapezoidal, and Clenshaw–Curtis methods, demonstrating the roles of numerical approximation accuracy and computational efficiency;
- demonstration of simultaneous identification of kernel structure and integration bounds through combined external and internal optimisation, showing that the framework enables joint discovery of both model form and parameters.

4.6.1 Limitations and future directions

Whilst this chapter presents a significant methodological advance in data-driven identification of distributed delay systems, several limitations merit acknowledgement. First, the current methodology assumes that delay values are known within reasonable bounds or can be estimated through external optimisation. Second, the framework requires adequate data resolution relative to system dynamics; systems with rapidly varying kernels may require dense sampling. Third, the curse of dimensionality in the parameter space poses practical challenges for systems with many unknown parameters, though particle swarm and Bayesian optimisation strategies substantially mitigate this limitation.

Regarding theoretical aspects, the formal convergence properties of the extended integral SINDy framework for distributed delays remain largely unexplored, representing an important direction for future theoretical investigation. Additionally, whilst the authors employ robust numerical approximation schemes, formal error analysis quantifying the relationship between quadrature errors and kernel reconstruction accuracy would strengthen the theoretical foundations.

Prospectively, the framework offers potential for substantial expansion in several directions. Integration of increasingly sophisticated and problem-specific candidate libraries tailored to particular application domains could enhance accuracy for systems with special structure. Application and validation of the methodology to real empirical data remains essential, requiring development of robust preprocessing techniques to handle measurement noise and missing observations. Hybrid estimation schemes combining statistical and machine learning formulations, particularly physics-informed neural networks for structure preservation, offer promising avenues. Development of adaptive quadrature strategies that optimise node placement based on kernel characteristics could improve efficiency. Extension to systems with multiple distributed delays and to coupled DIDE-RE systems mirrors the progression from discrete single delays to multiple discrete delays in prior chapters.

The following chapter applies the methodology to a real-world epidemiological system, demonstrating its ability to discover parsimonious distributed delay models directly from epidemiological time series. The application concerns the transmission dynamics of severe fever with thrombocytopenia syndrome (SFTS), a tick-borne disease, where the methodology identifies distributed delay structures representing the incubation period and other temporal dependencies in the transmission process. This application validates the practical utility of the framework and demonstrates its capacity to extract biological insight from observational epidemic data, paving the way for application to other disease systems exhibiting complex temporal dynamics and distributed memory effects.

5

Application of SINDy for a Tick-Borne disease

Contents

5.1 Introduction

The previous chapter has developed sparse identification methods for DIDEs and REs, establishing frameworks for discovering governing equations with distributed time delays from observational data. This chapter extends this approach to a practical and realistic epidemiological application: tick-borne disease transmission dynamics in a natural population structure. Specifically, we apply the SINDy framework to Severe Fever with Thrombocytopenia Syndrome (SFTS) [143, 147, 148], a tick-borne zoonotic disease in Dalian Province, China Data link, to demonstrate how distributed delay models can be identified directly from time-series data without requiring detailed a priori knowledge of transmission mechanisms. This chapter is based on the submitted work [29].

Tick-borne diseases represent a significant and growing public health challenge globally, with transmission dynamics heavily influenced by structured populations of vectors, hosts, and reservoir species. Traditional compartmental epidemiological models require extensive prior knowledge of population parameters, contact rates, and transmission pathways information that is often unavailable or uncertain in field settings. The SINDy methodology offers an alternative approach: by treating observed case incidence as a nonlinear dynamical system governed by distributed delays reflecting the age structure and maturation periods of tick populations, the method can automatically discover sparse representations of transmission dynamics directly from data.

The case study presented in this chapter focuses on SFTS transmission in Dalian, China, where detailed temporal surveillance data is available alongside environmental covariates such as temperature, which modulates vector activity and transmission rates. The tick vectors involved exhibit different life stages eggs, larvae, nymphs, and adults each with different infection dynamics and host-seeking behaviour. The structured population of ticks creates a complex history dependence in disease transmission: human cases depend not just on the current infected tick population but on accumulated exposures through the recent history of tick populations at different developmental stages. This memory effect manifests as distributed delays in the observed case incidence, making it an ideal candidate for application of the distributed delay SINDy (DD-SINDy) framework developed in the previous chapter.

5.1.1 Problem statement

The identification of SFTS transmission dynamics from observational data encounters unique challenges that distinguish this real-world application from controlled numerical experiments. First, the system involves multiple interacting structured populations (tick life stages, human infections, environmental drivers) whose dynamics are only partially observed, with only case incidence directly measured. Second, environmental factors such as temperature directly modulate transmission through nonlinear mechanisms that must be discovered from data. Third, the epidemiological data are subject to reporting delays, measurement error, and potential underestimation of true incidence. Fourth, the optimal form of the distributed delay kernel reflecting how past tick populations influence current human infections is unknown and must be estimated from limited time-series data.

The specific challenges addressed in this chapter include:

- identification of temperature-dependent transmission rates from observed case incidence;
- discovery of distributed delay kernels reflecting the influence of structured tick populations on human infections;
- validation of identified models through prediction accuracy and biological plausibility.

5.1.2 Motivation for distributed delay approach

Traditional deterministic compartmental models (such as SEIR-type systems [147]) treat transmission as depending only on the current state of the system. However, the life cycle of *Haemaphysalis longicornis* ticks, the primary vector for SFTS in the region, spans several months, during which the ticks pass through multiple developmental stages. At each stage, the probability of infection and transmission behaviour differs significantly. Human infections observed at time t thus depend not only on infected ticks present at time t, but on the accumulated history of infected ticks at various developmental stages throughout the preceding months. This history dependence defines a distributed delay.

Environmental temperature modulates this history dependence by controlling tick development rates and questing behaviour (host-seeking activity). Warmer temperatures accelerate tick development and increase questing activity, effectively changing the relevant time window over which past tick populations influence current infections. The SINDy methodology, combined with distributed delay formulations, allows automatic discovery of these temperature-modulated kernels directly from data.

5.1.3 Aim and objectives

The goal of this chapter is to demonstrate application of the distributed delay SINDy framework to real epidemiological data, specifically SFTS surveillance records in Dalian Province. The chapter illustrates how data-driven methods can discover simplified yet predictive representations of complex structured population dynamics in disease transmission.

The main objectives of this chapter are therefore as follows:

- to formulate SFTS transmission dynamics in Dalian as a distributed delay system incorporating structured tick populations and temperature-dependent transmission rates;
- to apply the SINDy framework to discover sparse distributed delay models directly from 12 years of monthly surveillance data (2011–2022) and environmental covariates;
- to compare predictions from identified SINDy models against observed case incidence, demonstrating the interpretability and predictive utility of the discovered sparse representations;
- to conduct sensitivity analyses on key identified parameters, particularly those governing the integration window (representing the relevant history of tick populations) and the temperature-dependent activity rates;
- to highlight the trade-off of data-driven and mechanistic approaches: the SINDy method sacrifices detailed biological interpretation for improved prediction accuracy and reduced prior knowledge requirements;
- to provide guidance for public health applications, including nowcasting of peak transmission periods and implications for vector control timing.

5.1.4 Chapter organisation

The chapter is organised as follows. Section 5.2 introduces the epidemiological background of SFTS, describing the disease characteristics, the vector species *Haemaphysalis longicornis*, and the ecological context of Dalian Province. Subsection 5.2.1 describes the surveillance data and environmental covariates used for model identification. Section 5.3

presents the application of the distributed delay SINDy methodology to discover two candidate models of increasing complexity: a simplified model using only temperature and case incidence, and a comprehensive model incorporating estimated infected tick populations at different life stages. Section 5.3.5 provides sensitivity analyses of key parameters to assess model robustness. Finally, Section 5.4 concludes the chapter, summarising key findings and implications for future applications of SINDy to epidemiological systems.

5.2 Severe fever with thrombocytopenia syndrome in Dalian

SFTS is an emerging disease transmitted from ticks to humans, first identified in central China in 2009. Since then, it has spread to South Korea, Japan, and other East Asian countries. The disease is caused by the SFTS virus (SFTSV), which belongs to the *genus Bandavirus* in the family *Phenuiviridae*.

Infected patients experience high fever, gastrointestinal symptoms (vomiting and diarrhoea), severe reductions in platelet count and white blood cell count, and elevated liver enzymes indicating liver damage. In severe cases, the disease can lead to failure of multiple organs, with a fatality rate of $12-50\%$ [65]. This high mortality rate makes SFTS a significant public health concern in endemic regions.

SFTSV can infect various animal species including rodents, dogs, and goats, which typically show no symptoms. These asymptomatic animals serve as viral reservoirs that maintain circulation in natural ecosystems. Humans become infected primarily through tick bites and human-to-human transmission can occur through direct contact with infected blood or body fluids [143].

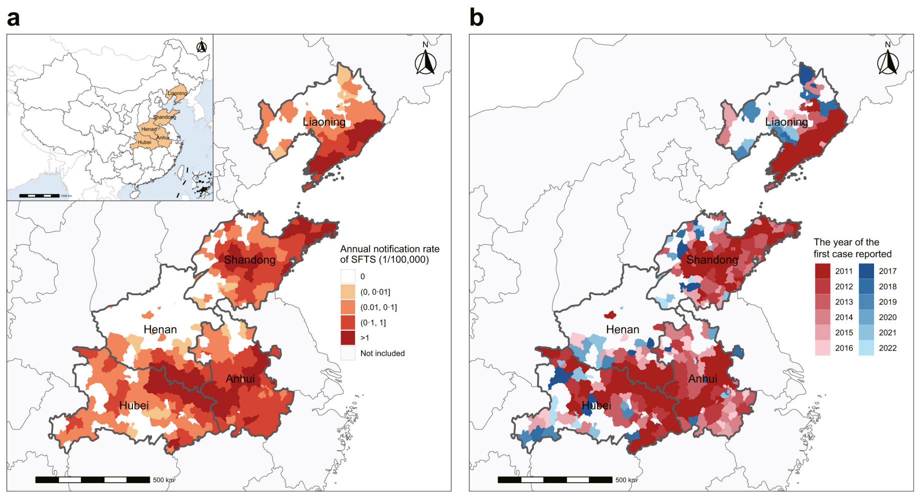

Figure 5.2.1: geographic distribution and spatial dynamics of reported SFTS notification rates by county in the five Chinese provinces with the highest notification rates from 2011 to 2022. (a) annual SFTS notification rate (1/100,000) from 2011 to 2022. (b) spatial dynamics of SFTS from 2011 to 2022. The colours in panel (b) indicate the first year within this period when cases were reported [56].

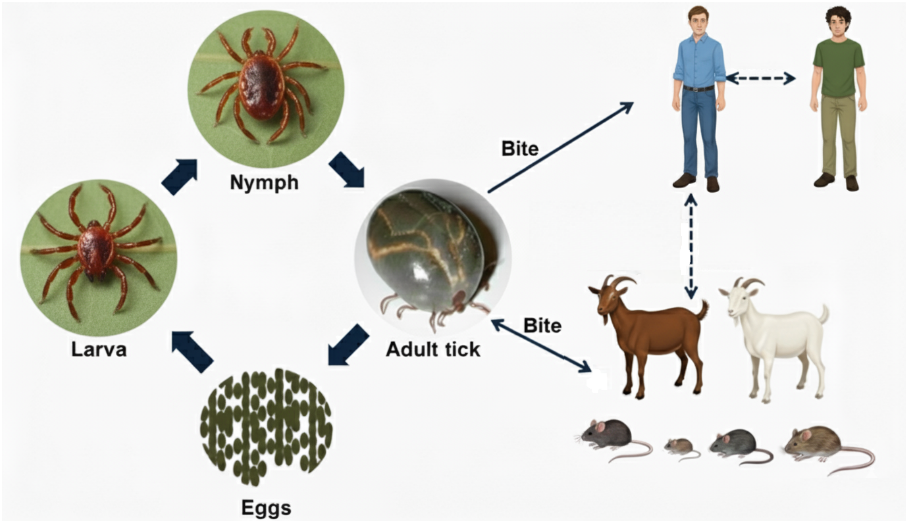

Figure 5.2.2: schematic transmission cycle of SFTSV highlighting the epidemiological pathways involving tick vectors, animal reservoirs, and human hosts. The illustration shows the main transmission route through bites from infected ticks to humans, together with the contribution of asymptomatic animal reservoirs in sustaining viral circulation and underpinning the zoonotic nature of SFTS transmission dynamics.

5.2.1 Data acquisition

This study is based on datasets that included environmental and epidemiological factors associated with SFTS in Dalian, Liaoning Province, China. Confirmed monthly cases of human SFTS reported from 2011 to 2022 were obtained from the Liaoning Province Center for Disease Control and Prevention, providing a general overview of the transmission dynamics at the community level. Currently, relevant climate data, specifically the monthly average temperatures recorded during this period, are sourced from the National Climatic Data Centre (NCDC), accessible through: Data link.

Figure 5.2.3 illustrates the seasonal comparison between temperature fluctuations and the incidence of SFTS, revealing a significant visual correlation that supports the hypothesis that climatic factors, particularly temperature, affect tick behaviour and human contact patterns. Additionally, the unique ecological context of Dalian, primarily marked by a high prevalence of *Haemaphysalis longicornis* ticks and specific host populations, such as nymph ticks associated with rodents and adult ticks found on goats, provides an ideal setting for unravelling the intricate factors that contribute to outbreaks of vector-borne diseases. In adherence to established protocols, these datasets are standardised through preprocessing and gaps arising from reporting inconsistencies are addressed. The structure complies with recognised best practices concerning climate data usage and time series analysis in modelling vector-borne diseases. Prior studies have highlighted the necessity of region-specific high-resolution datasets and emphasised the importance of ensuring temporal alignment between epidemiological and environmental variables [148].

All datasets were aligned to ensure that epidemiological and environmental variables were temporally synchronised for model fitting [148]. This curated dataset enables the construction of high-dimensional, temporally aligned feature matrices suitable for training and validating SINDy-based models.

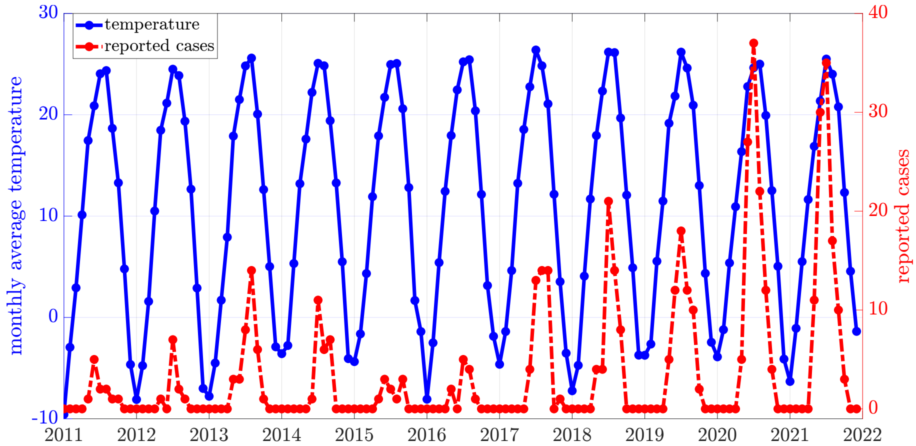

Figure 5.2.3: temporal relationship between environmental and epidemiological factors in Dalian from 2011 to 2022. The blue solid line depicts the monthly average ambient temperature, whereas the red dashed line shows the monthly count of confirmed SFTS cases.

5.3 SINDy-based distributed delay model discovery

5.3.1 Mechanistic baseline model

The reference paper [148] develops a mechanistic, stage-structured compartmental model formulated as a system of ODEs to understand SFTS transmission dynamics in Dalian. The ODE system comprises sixteen state variables representing tick populations at different life stages and infection statuses, plus rodent and goat host populations, yielding a coupled system of twenty DEs for the complete tick host virus dynamics. This model explicitly incorporates the complete life cycle of the primary vector, Haemaphysalis longicornis, tick populations across eight distinct life stages: eggs, questing larvae, feeding larvae, questing nymphs, feeding nymphs, questing adults, feeding adults, and egg-laying adults. Each of these life stages is further subdivided into subcategories (susceptible or infected), recognising that different developmental stages possess distinct infection dynamics, host seeking behaviour, and transmissibility. This age-structured approach is biologically motivated: immature ticks (larvae and nymphs) predominantly feed on rodents, while adult ticks preferentially parasitise larger hosts such as goats, creating distinct ecological roles and transmission opportunities that a simplified, age-structured model would necessarily overlook.

A distinctive feature of the mechanistic model is its explicit treatment of three coexisting transmission pathways through which SFTSV circulates within the tick-host system. Systemic transmission occurs when susceptible ticks acquire infection while feeding on infected animal hosts (or conversely, when infected ticks transmit virus to susceptible hosts during feeding). Co-feeding transmission is a non-systemic route wherein susceptible and infected ticks simultaneously feed on the same host, enabling virus transmission through host mediated mechanisms. Transovarial transmission represents vertical transmission of SFTSV from infected adult female ticks to their offspring via eggs, providing a mechanism for virus persistence even in the absence of host infections.

The mechanistic model encodes the temperature dependent nature of SFTS transmission by making virtually all biological rates functions of ambient temperature: tick development rates from egg to larva to nymph to adult, tick mortality rates at each life stage, and tick host seeking activity rates all follow temperature-dependent functional forms fitted to experimental data. This temperature dependence is critical because tick life cycle directly governs the structure and magnitude of the infected tick populations available to transmit virus to humans. Additionally, the model incorporates logistic birth dynamics for the primary host populations (rodents and goats), accounts for natural mortality at each stage, and includes a one month diagnostic reporting delay to realistically represent the lag between tick exposure and confirmed SFTS diagnosis in humans.

The crucial quantity connecting tick host dynamics to observable human cases is the monthly case incidence, which is computed through an integral equation that aggregates the contributions of infected questing nymphs and infected questing adults over a one month surveillance period, weighted by their respective temperature modulated attachment rates to humans. Specifically, the predicted number of newly reported human SFTS cases during month t is expressed as:

$$\hat{y}(t) = \int_0^{\sigma} \left(\Lambda_N(t-a)p_N(t-a)N_{qi}(t-a) + \Lambda_A(t-a)p_A(t-a)A_{qi}(t-a)\right) \mathrm{d}a, \quad (5.3.1)$$

where σ denotes the surveillance interval, N_{qi} and A_{qi} represent the densities of infected questing nymphs and infected questing adults at time, respectively, and Λ_N and Λ_A are temperature dependent host attachment rates for nymphs and adults. These attachment rates are formulated as $\Lambda_\beta = \alpha_\beta e^{\omega T}$, where α_β represents temperature-dependent strength coefficients, ω encodes the strength of human outdoor activity, and T denotes ambient temperature. This formulation represents the key biological insight that human case incidence depends not merely on the current state of the system but rather on the accumulated history of infected tick populations weighted by their behavioural activity, a concept intimately connected to the distributed delay framework employed in the subsequent SINDy based analysis.

5.3.2 SINDy-based model discovery

Building upon the theoretical framework developed in Chapter 4 for extending SINDy to distributed delay systems with nonautonomous kernels, this section demonstrates its practical applicability to identify a nonautonomous kernel of a distributed delay model based on multivariate observational data aimed at forecasting SFTS transmission dynamics in Dalian Province. Rather than building a mechanistic model from first principles, our approach directly identifies the underlying dynamical relationships from observed time series data. The dataset includes four key variables: human case counts, environmental temperature measurements, and abundance data for two tick life stages (nymphs and adults). The tick population data was derived from a transmission dynamics model [148]. By integrating these complementary data streams within the SINDy framework, we capture the complex relationships between tick life-stage dynamics, environmental conditions, and human infection incidence. This approach reveals the sparse structure of the underlying distributed delay kernel, which enables us to forecast SFTS epidemiological patterns effectively.

The monthly records of SFTS cases from Dalian spanning 2011 to 2022, along with daily temperature data, serve as input for model identification. The data is divided into training and validation subsets, with the first eight years (96 data points) dedicated to model training and parameter estimation, while the remaining years are set aside for validation and forecasting evaluation. Polynomial SINDy libraries were constructed in combination with trapezoidal quadrature $K = 100$ for all the experiments. In this study, we set λ to be very low ($\lambda = 10^{-10}$) to ensure the coefficients match with the values found in the reference model [148]. This choice prioritises biological interpretability over maximum sparsity, avoiding the removal of terms just to obtain a more sparse model.

5.3.3 Model evolution case: 1

A library of potential kernel functions is created, beginning with terms that include human cases (denoted as C), temperature data (denoted as T) and their combinations (5.3.2). Next, an exponential term representing the impact of outdoor activity, expressed as $e^{\omega T}$, is added to the database (5.3.3). Additionally, a model to account for the distributed memory effect is constructed using numerical quadrature, where the time-delay interval is discretised to establish the regression matrix, and sparse regression is employed to identify the minimal set of functional terms needed to reflect the time-series behaviour observed in prior cases.

Two alternative model formulations are investigated. The first model takes the form

$$\hat{y}(t) = \int_0^{\sigma} f(C(t-a), T(t-a))\,\mathrm{d}a, \tag{5.3.2}$$

whilst the second incorporates the outdoor activity term s.t.,

$$\hat{y}(t) = \int_0^{\sigma} f(C(t-a), T(t-a), e^{\omega T(t-a)})\,\mathrm{d}a \tag{5.3.3}$$

where $\hat{y}$ denotes the newly reported cases of human SFTS during a specific time period σ, with the reporting period set to one month. The parameter ω is externally optimised by applying particle swarm optimisation (PS). Data splitting is conducted to assess the generalisability of the model.

Figure 5.3.1 shows a comparison between the actual reported human cases and the predictions made by the SINDy model, which employs two different libraries with varying polynomial orders using human case data and temperature data (5.3.2). The figure demonstrates the capacity of the model to capture the temporal dynamics associated with the incidence of SFTS, with predictions closely matching the observed case counts during both the training and validation phases. The visual comparison indicates that both polynomial orders provide reasonable approximations, although performance differences are noticeable across various time intervals.

Figure 5.3.2 illustrates a comparison between observed human case data and the predictions made by the SINDy model when an exponential term (5.3.3) is incorporated into the library for different polynomial orders. This exponential term accounts for the influence of temperature on outdoor activities. The findings indicate that adding this term results in predictive performance similar to the baseline model (5.3.2), with a polynomial order of one showing significantly better alignment between predictions and observed cases than higher-order polynomial terms.

	ξ (degree = 1)		ξ (degree = 2)	
library terms	(5.3.2)	(5.3.3)	(5.3.2)	(5.3.3)
C	3.04×10^{-2}	3.26×10^{-2}	6.95×10^{-2}	6.20×10^{-2}
T	1.77×10^{-4}	1.66×10^{-4}	1.35×10^{-4}	2.49×10^{-4}
$e^{\omega T}$	-	-	-	1.85×10^{-9}
C^2	-	-	2.33×10^{-2}	2.09×10^{-4}
$C \cdot T$	-	-	-1.66×10^{-3}	-1.35×10^{-3}
T^2	-	-	-7.48×10^{-6}	-1.57×10^{-5}
RMSE				
training	1.67×10^{0}	1.63×10^{0}	1.62×10^{0}	1.61×10^{0}
validation	4.96×10^{0}	4.99×10^{0}	1.20×10^{1}	1.04×10^{1}

Table 5.3.1: sparse coefficient matrices for the distributed delay models defined in (5.3.2) and (5.3.3), identified using the STLS regression approach with polynomial degrees one and two. The table reports the nonzero coefficients associated with the state variables C (human cases), T (temperature), and $e^{\omega T}$ (exponential term representing outdoor activity), together with their nonlinear interactions. RMSE values are given for both the training and validation datasets.

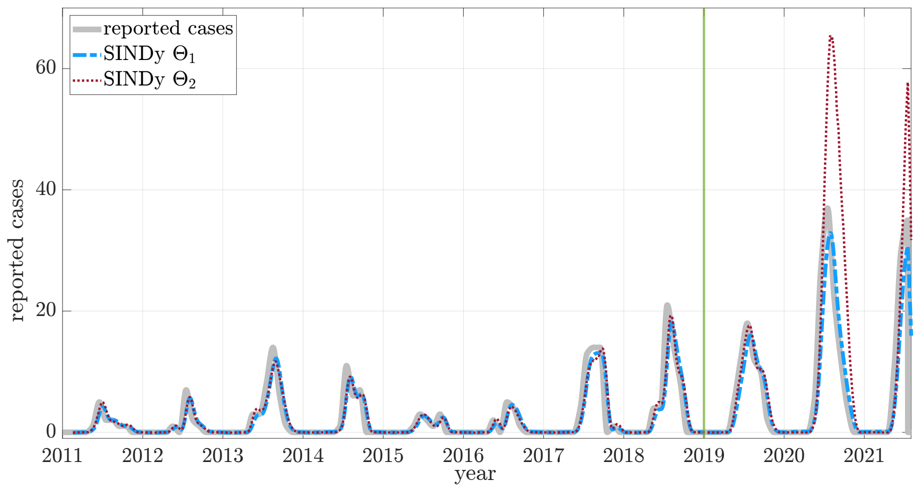

Figure 5.3.1: predictive performance of the DD-SINDy model for SFTS cases using two distinct polynomial-order libraries within the candidate function set (5.3.2). The solid grey line denotes the actual number of human cases, while the blue and red lines show SINDy predictions obtained with polynomial degrees one and two, respectively.

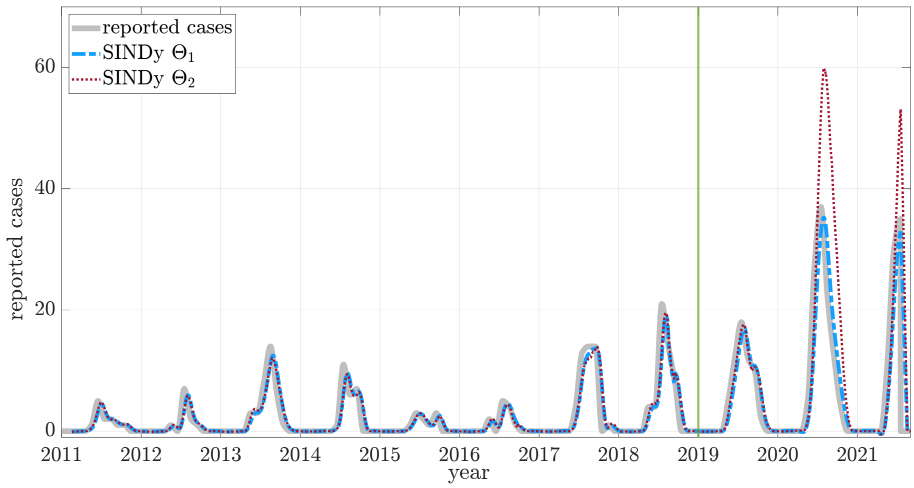

Figure 5.3.2: predictive performance of the DD-SINDy model for SFTS cases using two distinct polynomial-order libraries within the candidate function set (5.3.3), including an exponential term $e^{\omega T}$ to represent temperature-dependent outdoor activity effects. The solid grey line denotes the actual number of human cases, while the blue and red lines show SINDy predictions obtained with polynomial degrees one and two, respectively.

5.3.4 Model evolution case: 2

Expanding on the results of the previous analysis, we extend the model by integrating data of infected tick populations with human case records and temperature data. This improved method takes advantage of the multivariate nature of the transmission system, acknowledging that the dynamics of SFTS are closely linked to the presence of infected vectors at various life stages. Data on infected nymph and adult tick populations are obtained from simulations of the mechanistic transmission model detailed in [148]. As in the previous section, we use the complete dataset from Dalian Province covering the years 2011 to 2022, applying the same temporal division, the first eight years for training the model and the subsequent period for validation. By incorporating these complementary data streams into the SINDy framework, the approach captures the complex interdependencies between tick population dynamics, environmental factors, and human infection rates, thus allowing the identification of a more comprehensive representation of the underlying distributed delay kernel. This relationship can be expressed through the following formulation

$$\hat{y}(t) = \int_0^{\sigma} f(C(t-a), T(t-a), e^{\omega T(t-a)}, I_{N_q}(t-a), I_{A_q}(t-a))\, \mathrm{d}a \tag{5.3.4}$$

where I_{N_q} and I_{A_q} represent infected nymph and adult tick populations, respectively.

Figure 5.3.3 compares actual human cases with SINDy predictions using data that include human cases, temperature, and infected nymph and adult tick populations in the library. The inclusion of these data appears to capture additional dynamics not represented by environmental and epidemiological variables alone.

The final identified model for the prediction of SFTS cases assumes the form of an integral equation, wherein newly predicted cases are expressed as a sparse linear combination of selected kernel terms, each integrated over the relevant delay windows. The

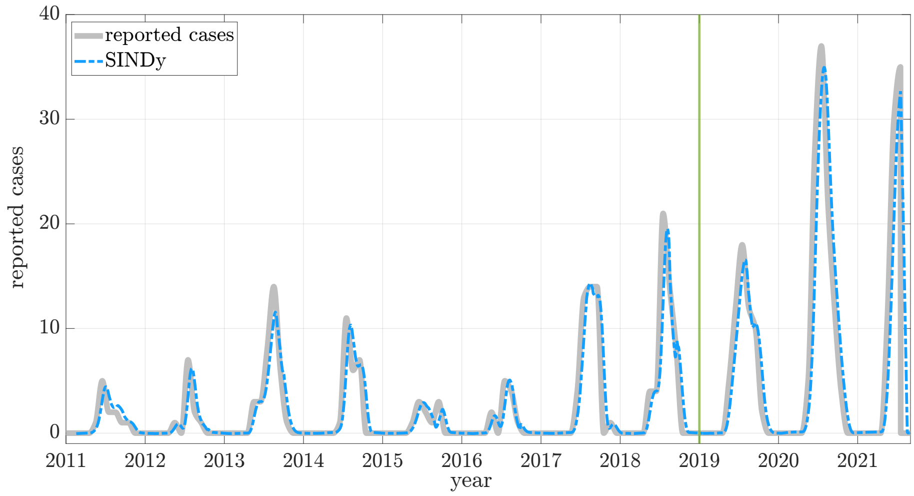

Figure 5.3.3: predictive performance of the DD-SINDy model for SFTS cases using polynomial-order library within the candidate function set infected nymph and adult populations, as well as environmental and human epidemiological factors (5.3.4). The solid grey line denotes the actual number of human cases, while the blue line shows SINDy prediction obtained with polynomial degree one.

library terms	C	T	$e^{\omega T}$	I_{N_q}	I_{A_q}
ξ	2.61×10^{-2}	-7.75×10^{-4}	3.55×10^{-8}	6.58×10^{-5}	-7.03×10^{-8}
RMSE	training			validation	
	1.51×10^{0}			4.88×10^{0}	

Table 5.3.2: sparse coefficient matrix for the distributed delay model (5.3.4), which includes human cases (C), temperature (T), exponential outdoor activity ($e^{\omega T}$), infected nymph populations (I_{N_q}), and infected adult tick populations (I_{A_q}) as state variables. Coefficients are recovered using STLS regression with a polynomial degree of one. The table reports RMSE values for both the training and validation phases.

structure of the discovered model exhibits a limited number of non-zero coefficients, consistent with the sparsity-promoting objective of the SINDy framework. The SINDy-based distributed delay model demonstrates the best fit to the training data sample, as indicated by the low RMSE values reported throughout the analysis.

Figure 5.3.4 presents the comparison of the trajectories produced by the mechanistic model [148] and SINDy discovered model (5.3.4) with real monthly reported human cases data.

5.3.5 Parameter sensitivity analysis

Sensitivity analysis plays a vital role in both validating and interpreting data-driven epidemiological models. Firstly, it measures robustness by evaluating how model predictions respond to changes in identified parameters. This helps determine if the parameters reflect true aspects of the transmission dynamics or are simply numerical errors from the optimisation process. Secondly, it enhances interpretability by identifying which param-

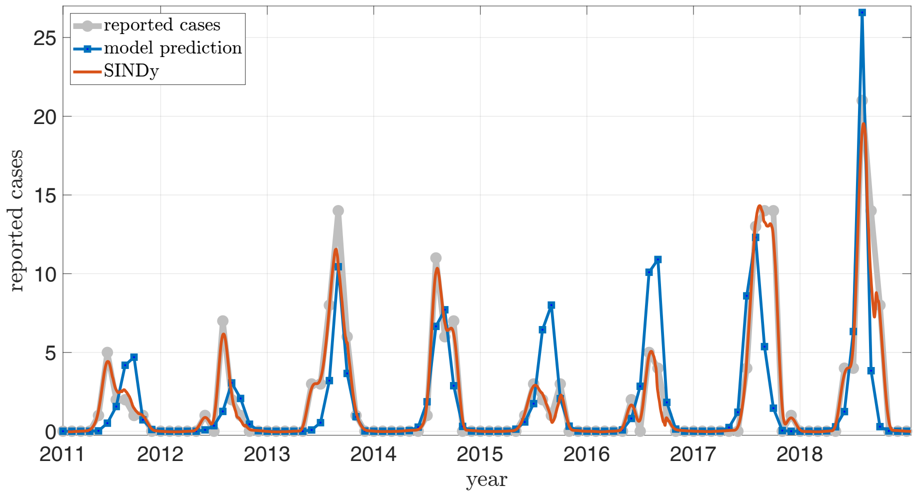

Figure 5.3.4: comparison of monthly reported cases (solid thick grey) with model predictions (solid blue) [148] and SINDy-based predictions (solid red) (5.3.4).

eters most significantly impact predictions, thereby highlighting the dominant epidemiological mechanisms in SFTS transmission in Dalian.

The DD-SINDy model identified includes two crucial parameters that require thorough sensitivity analysis. The outdoor activity parameter ω (in the exponential term $e^{\omega T}$) represents how temperature nonlinearly affects human exposure to infected ticks through behavioural changes, with warmer temperatures leading to more outdoor activity and higher contact rates. The integration window parameter σ (defining the reporting period) signifies the relevant time frame over which past environmental conditions and vector populations affect current human infections.

Biologically, both parameters are expected to be sensitive. The parameter ω should fall within a physiologically reasonable range, as extreme values would suggest either minimal or unrealistically high temperature dependence. The parameter σ should align with the actual tick life cycle duration in the climate of the Dalian and deviations from this window should reduce prediction accuracy if the identified value truly captures the system's intrinsic timescale rather than being a fitting error.

The outdoor activity parameter ω is selected externally via PSO on a validation set, and ω_{opt} denotes the value minimising the validation RMSE used for model selection. Figures 5.3.5 and 5.3.6 instead, depict a local sensitivity analysis obtained by perturbing ω around ω_{opt} while keeping the remaining model structure fixed, which explains why the plotted RMSE is not strictly minimal at ω_{opt}. The reporting window $\sigma = 1$ month is not tuned but imposed from epidemiological practice and previous SFTS studies, which operate on monthly case counts; the corresponding RMSE curve is therefore used only to confirm that this literature-based choice lies in a stable region of the error landscape rather than at a sharp optimum.

In this section, we systematically adjust these parameters across feasible ranges and measure the resulting changes in validation error metrics.

Outdoor activity parameter ω

The exponential human activity $e^{\omega T}$ modulates the temperature dependent contact rate between humans and infected tick populations. This term captures the relationship between ambient temperature and outdoor recreational activities that increase human exposure to tick-borne pathogens.

We evaluate the performance of the model over a range of ω values.

$$\omega \in \left[0.1 \cdot \omega_{\text{opt}}, 1\right] \tag{5.3.5}$$

where ω_{opt} denotes the optimal value obtained through particle swarm optimisation (PS). For each candidate value ω_i on a discretised grid of 15 points, we reconstruct the SINDy model using Algorithm A.8 and compute the validation metrics. The corresponding validation error is computed as:

$$\epsilon(\omega_i) = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j(\omega_i)\right)^2}. \tag{5.3.6}$$

Deviation from ω_{opt} by more than $\pm 30\%$ results in significant increases in RMSE, demonstrating the critical role of the parameter in capturing transmission dynamics.

Integration window parameter σ

The parameter σ defines the temporal extent of the distributed delay kernel, representing the epidemiologically relevant time window over which past environmental conditions and tick populations influence the current incidence of SFTS. We extend the analysis range compared to previous studies to explore potential long-term memory effects:

$$\sigma \in [0.25, 3.0] \text{ months.} \tag{5.3.7}$$

For each σ_k, the distributed delay integral is approximated through quadrature method.

$$\hat{y}(t; \sigma_k) = \int_0^{\sigma_k} f(C(t-a), T(t-a), e^{\omega T(t-a)}, I_{N_q}(t-a), I_{A_q}(t-a))\, \mathrm{d}a. \tag{5.3.8}$$

Figure 5.3.5 presents a parameter sensitivity analysis examining the effects of the outdoor activity parameter ω and the human case reporting period σ. The sensitivity analysis provides a quantitative assessment of how variations in these parameters influence the performance of the model and the accuracy of the prediction.

Figure 5.3.6 displays the parameter sensitivity analysis of the outdoor activity parameter and the human case reporting period while employing data on human cases, temperature, and infected nymph and adult tick populations. The sensitivity profiles provide a comprehensive assessment of the influence of the parameters across the expanded model space, allowing the identification of parameters that have the greatest impact on prediction accuracy.

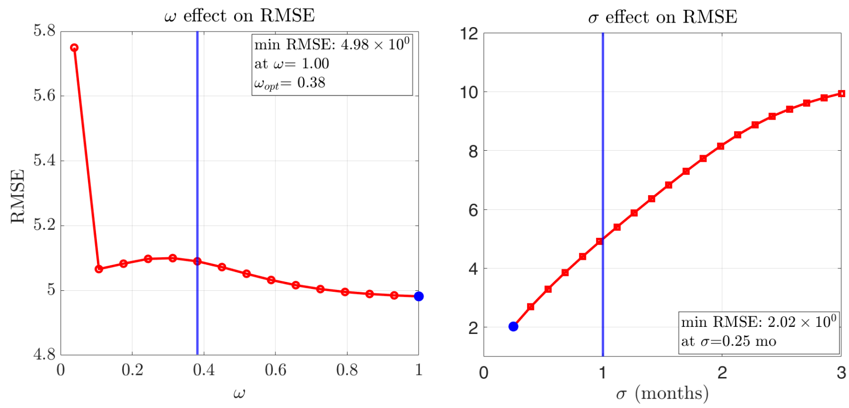

Figure 5.3.5: sensitivity analysis was conducted to evaluate how resilient the model predictions are to changes in two critical parameters (5.3.3): the outdoor activity coefficient ω and the integration window parameter for the human case reporting period σ, while also including the validation RMSE across a discretised range of parameter values, vertical blue line in the ω panel represents the optimal value obtained during training phase and vertical blue line in σ panel represents the one month human cases reporting period.

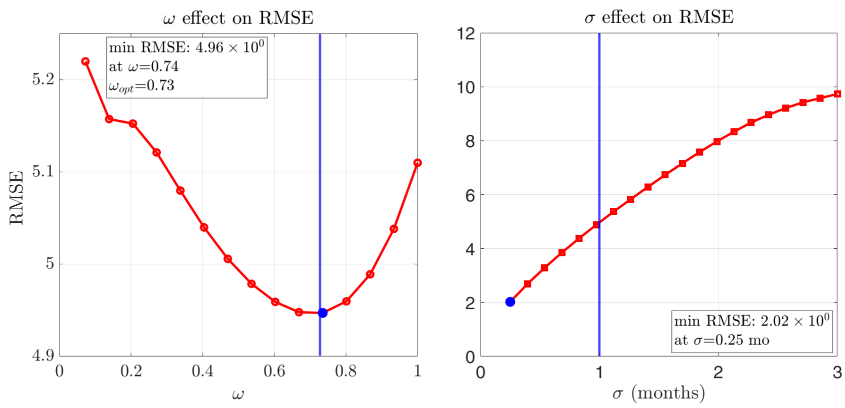

Figure 5.3.6: extended parameter sensitivity analysis for the comprehensive distributed delay model incorporates tick population data, which includes both infected nymphs and adults, along with human cases and temperature (5.3.4). This analysis also includes the validation RMSE across different combinations of the outdoor activity parameter ω and the integration window parameter for the human case reporting period σ, vertical blue line in the ω panel represents the optimal value obtained during training phase and vertical blue line in σ panel represents the one month human cases reporting period.

5.4 Conclusions

This chapter demonstrates a practical application of the DD-SINDy framework to real epidemiological data, specifically SFTS transmission dynamics in Dalian Province, China. The study shows that data-driven identification of sparse distributed delay models provides complementary insights to traditional mechanistic epidemiological approaches.

The identified SINDy-based distributed delay models reveal several important aspects of SFTS transmission dynamics

- the discovered models confirm relationships between ambient temperature and transmission rates, captured through temperature-modulated activity terms. These relationships are consistent with known biology of tick development and questing behaviour but are discovered automatically from data without prior specification;
- the identified integration windows (σ) reflect the life cycle duration of tick vectors in the region, providing a mechanistic interpretation of the dynamics of the structured population. This memory window captures the cumulative effect of tick populations at different developmental stages on current human infections;
- both identified models achieve superior prediction performance compared to the mechanistic baseline model when evaluated on validation data;
- sensitivity analyses demonstrate that identified models are robust to moderate parameter variations, though prediction accuracy degrades significantly beyond this range, indicating that the identified parameter values represent genuine features of the underlying dynamics rather than artefacts of the optimisation procedure.

Although mechanistic compartmental models such as [148] provide valuable biological insight through explicit representation of tick life stages, reservoir dynamics, and basic reproduction numbers (R_0), our SINDy method presents complementary advantages:

- SINDy does not require extensive prior assumptions about model structure, transmission mechanisms, or parameter values. The method discovers governing equations automatically from data;
- the sparse regression procedure automatically uncovers interactions and functional forms (such as temperature-modulated contact rates) that might be overlooked or difficult to justify *a priori*;
- SINDy models are directly optimised for prediction accuracy rather than for reproducing known biological parameters, making them particularly suitable for public health applications such as nowcasting and forecasting;
- the identified sparse models are substantially simpler than comprehensive compartmental models, enabling rapid implementation and evaluation in surveillance settings.

However, this comes with recognised trade-offs:

- SINDy-identified models do not explicitly track all biological compartments and cannot directly compute classical epidemiological metrics such as R_0 or generation times;

- distributed delay formulation averages temporal variations through kernel functions, potentially obscuring fine-scale biological details captured by age-structured compartmental models;
- models discovered from observed dynamics may not extrapolate reliably to conditions outside the training data range (e.g., climate change scenarios), whereas mechanistic models grounded in biological principles may be more robust to extrapolation.

5.4.1 Limitations and future directions

Several limitations of the current study suggest directions for future work:

- the analysis relies on reported SFTS cases, which may underestimate true incidence due to underreporting, particularly in rural areas;
- model (5.3.4) relies on estimated infected tick population data rather than direct field collections;
- the current analysis emphasises temperature as the primary environmental driver;
- the current analysis focuses on a single region;
- longer-term prospective validation on future seasons would strengthen confidence in model reliability for operational forecasting applications.

This chapter concludes the demonstration of the DD-SINDy framework applied to real epidemiological systems. The next chapter addresses a fundamental question: **how does the data-driven discovery approach compare to other contemporary methods for system identification from time-series data?** Specifically, the following chapter compares the sparse, interpretable models discovered through SINDy with neural network based approaches. This comparison examines the trade-offs between model interpretability and flexibility, predictive accuracy, data efficiency, and computational requirements across multiple benchmark problems. Such comparative analysis is essential for practitioners choosing appropriate methods for their specific applications and for understanding the complementary roles of mechanistic, sparse, and deep learning approaches in system identification.

6

Comparative analysis with neural network approaches

Contents

6.1 Introduction

This chapter synthesises the developments presented in the preceding chapters and provides a discussion of the sparse identification methodologies developed for delay differential equations (DDEs) (E-SINDy and P-SINDy) with the neural delay differential equations (NDDEs). Building upon the foundations established in Chapter 1 and the detailed methodology in Chapter 2 that introduced E-SINDy and P-SINDy, this chapter accomplishes two primary objectives. First, it places the sparse identification approach within the broader landscape of data-driven system identification by presenting a comparative analysis with neural network-based methods. Second, it discusses the key findings, limitations, and future research directions that emerge from this comprehensive investigation. This chapter is primarily based on the framework discussed in the book chapter [25], from which all figures have been reproduced.

The thesis has developed several complementary sparse identification frameworks. The E-SINDy method extends the standard SINDy framework to DDEs when delay values are known or can be estimated, providing direct interpretability of governing equations through sparse regression. The P-SINDy method employs pseudospectral collocation to reformulate the infinite-dimensional DDE problem as a finite-dimensional ODE system, requiring optimisation of only the maximum delay and offering improved computational efficiency at the cost of reduced interpretability. This E-SINDy framework has been extended also to SDDEs and distributed delay systems.

As a reference point for comparison, this chapter also discusses NDDEs, a powerful black-box learning approach that has emerged recently in the machine learning literature [74]. Although NDDEs are not the primary focus of this thesis, understanding their capabilities, limitations, and relationship to sparse identification methods provides valuable context for practitioners to select appropriate identification tools for specific applications.

This chapter is organised as follows. Section 6.2 presents an overview of the E-SINDy and P-SINDy with NDDEs approaches for DDE identification. Section 6.3 provides a detailed comparative analysis of E-SINDy, P-SINDy, and NDDE methods. Finally, Section 6.4 concludes the chapter.

6.2 Overview of data-driven approaches for DDEs identification

Data-driven identification of DDEs has emerged as a central paradigm in many scientific fields. The fundamental challenge is to discover both the functional form of the governing equations and the delay parameters directly from time-series measurements.

6.2.1 Sparse identification approaches (E-SINDy and P-SINDy)

The SINDy framework is based on the premise that governing equations can be represented as sparse linear combinations of candidate functions. Within the scope of DDEs, this thesis explores two distinct implementations:

E-SINDy maintains the precise structure of the DDE by directly integrating delayed variables into the library, resulting in highly interpretable coefficients that align with physical mechanisms. However, its computational demands increase significantly with the number of unknown delays, as it necessitates a multivariate external search (via [116], BO [104], or PS [16]) across the delay parameter space.

P-SINDy focuses on computational efficiency by transforming the DDE into a finite-dimensional ODE system using pseudospectral collocation. This approach simplifies the optimisation problem to a search for a single parameter (the maximum delay $\bar{\tau}$), irrespective of the number of delays present in the system. The downside is a decrease in direct interpretability, as the explicit delay structure is masked by the collocation state variables.

6.2.2 Neural delay differential equations

NDDEs represent an alternative paradigm where neural networks with trainable delay parameters learn the RHS of DDEs through gradient-based optimisation [74]. The key distinguishing features are:

- neural networks can approximate arbitrary nonlinearities without prior specification of functional forms, offering flexibility for systems with unknown structure.
- delays and network weights are optimised jointly through backpropagation, avoiding nested optimisation loops required by E-SINDy.

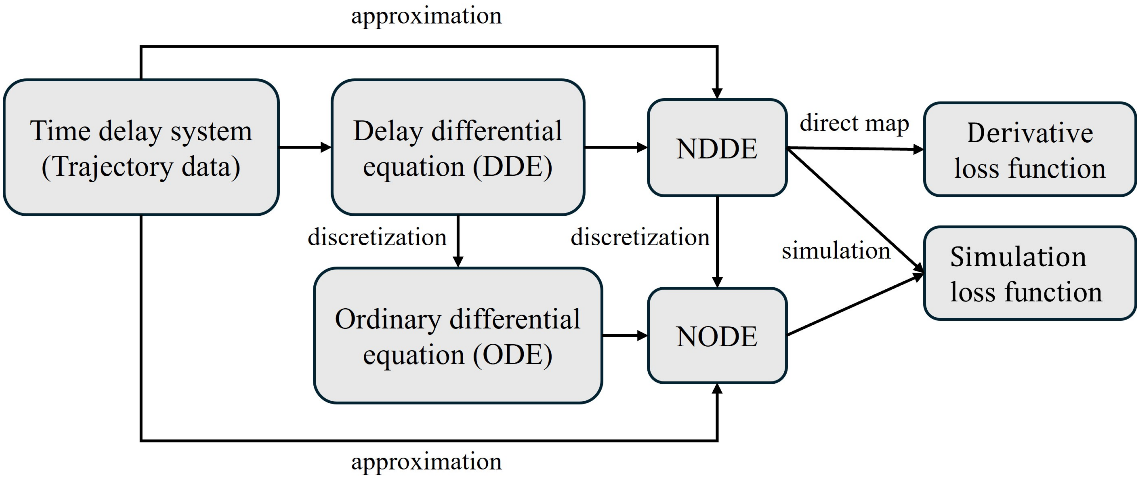

Figure 6.2.1: overview of modelling time delay systems with neural networks and differential equations [25].

- neural networks typically require larger datasets and more computational resources for training compared to sparse methods but leverage multiple trajectories to improve performance.
- while explicitly trainable delays provide some structure, the millions of parameters in deep networks reduce overall model interpretability compared to sparse linear combinations of basis functions.

Formulating the neural delay differential equations

System (1.2.1) with k discrete delays, $0 = \tau_0 \leq \tau_1 < \cdots < \tau_k \leq \tau_{\max}$, can be rewritten as

$$x'(t) = f(\mathbf{x}_t), \tag{6.2.1}$$

where $\mathbf{x}_t = [x(t), x(t-\tau_1), \ldots, x(t-\tau_k)]^\top \in \mathbb{R}^{n(k+1)}$ and $f : \mathbb{R}^{n(k+1)} \to \mathbb{R}^n$. In order to capture the dynamics governed by (6.2.1), we construct the NDDE analogously:

$$\hat{x}'(t) = \text{net}(\hat{\mathbf{x}}_t), \tag{6.2.2}$$

with $k+1$ trainable delays, $0 \leq \hat{\tau}_0 < \hat{\tau}_1 < \cdots < \hat{\tau}_k \leq \tau_{\max}$, and net : $\mathbb{R}^{n(k+1)} \to \mathbb{R}^n$ representing the RHS through a time delay neural network. Suppose the network has L layers with nonlinearity $g_l(\cdot)$ in each layer $l = 1, \ldots, L$. Let us denote the input as $\mathbf{z}_0^t$ and the output as z_L^t at time t. Then we have

$$z_L^t = \text{net}(\mathbf{z}_0^t), \tag{6.2.3}$$

where

$$\begin{aligned} z_1^t &= g_1(W_1 \mathbf{z}_0^t + b_1), \\ z_l^t &= g_l\big(W_l z_{l-1}^t + b_l\big), \quad l = 2, \ldots, L, \end{aligned} \tag{6.2.4}$$

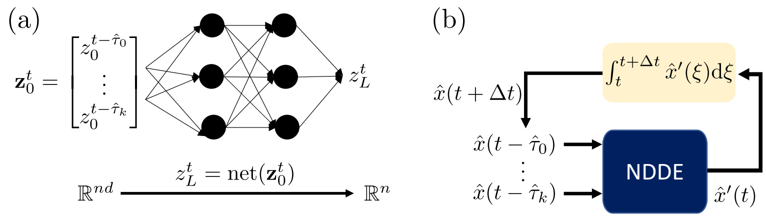

Figure 6.2.2: (a) illustration of a time delay neural network constructed to capture the dynamics of a neural delay differential equation (NDDE) with a single delay. (b) illustration of the simulation of an NDDE [25].

where W_l and b_l denote the weights and biases, respectively, and one may use different nonlinearities g_l including the rectified linear unit (ReLU) [53, 68, 91] and different sigmoidal functions [63]. This network can be used to approximate the state derivative of the delayed dynamical system by defining $z_L^t = \hat{x}'(t)$ and $\mathbf{z}_0^t = \hat{\mathbf{x}}_t$, cf. (6.2.2) and (6.2.3).

Note that some of the delays trained in a NDDE do not have to be in an ascending/descending order and they can potentially have the same value. When the NDDE (6.2.2) is used to approximate the dynamics of the time delay system (6.2.1), knowing the number of delays in the system is beneficial but not required [74]. Moreover, if the existence of a non-delayed term with $\tau_0 = 0$ is known, one may fix the delay value $\hat{\tau}_0 = 0$ to further restrict the neural network. Figure 6.2.2(a) illustrates a time delay neural network used to capture the RHS of an NDDE with discrete delays, while panel (b) shows a block diagram illustrating the corresponding NDDE simulation. We emphasise that below we construct time delay neural networks with trainable delays, that is, apart from the weights W_l and biases b_l, $l = 1, \ldots, L$ in (6.2.4), we also aim to learn the delays $\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_k$ themselves.

For example, an autonomous delay dynamical system with one delay ($k = 1$) can be described by

$$\dot{x}(t) = f(x(t), x(t-\tau)), \tag{6.2.5}$$

whose RHS f and delay τ may be unknown. Our goal is to learn f as well as τ from the data using the NDDE

$$\hat{x}'(t) = \mathrm{net}(\hat{x}(t), \hat{x}(t-\hat{\tau})), \tag{6.2.6}$$

with trainable delay $\hat{\tau}$.

Note that control systems with state feedback can also be viewed as autonomous systems. For example, with slight abuse of notation, the system $\dot{x}(t) = f(x(t), u(t))$ with delayed state feedback $u(t) = k(x(t-\tau))$ yields

$$x'(t) = f\big(x(t), k(x(t-\tau))\big). \tag{6.2.7}$$

When the nonlinear dynamics f of the system, the feedback law k, and the delay τ are not known, we need to learn these from trajectory data.

The trained neural network (6.2.6) can then be solved by any DDE solver for a given initial history. The state at time t is given by

$$\hat{x}(t) = \mathrm{DDEsolver}(\mathrm{net}, x_{t_0}, t), \tag{6.2.8}$$

where the history function $x_{t_0}(s), s \in [t_0 - \tau_{\max}, t_0]$ is approximated from data $x(t_0), x(t_0 - \Delta t), \ldots, x(t_0 - \tau_{\max})$ and Δt denotes the sampling time.

Due to the trainable delays in the networks and the recurrent feature of network simulation, the time delay neural network is able to take the time dependency into consideration while keeping the number of parameters relatively low. Moreover, once the dynamics of the DDE (6.2.2) are captured by the neural network (6.2.3), they can be analysed using tools from the literature on dynamical systems and control. Additionally, with explicit delay parameters in the input layer, the network is more interpretable and generalisable.

6.3 Comparison of SINDy and NDDEs approaches

In this section, we evaluate the SINDy approach and the NDDE approach on the 3-state delayed Rössler system. The reconstructions of the time delays and the trajectories, as well as the computation time, are compared between different models and training methods. In addition, the advantages and shortcomings of the different approaches are also discussed in the section.

Example 6.3: Delayed *Rössler system*

The double delayed *Rössler system* [58, 144, 146] is given by

$$\begin{cases} x_1'(t) = -x_2(t) - x_3(t) + \alpha_1 x_1(t-\tau_1) + \alpha_2 x_1(t-\tau_2), \\ x_2'(t) = x_1(t) + \beta_1 x_2(t), \\ x_3'(t) = \beta_2 + x_3(t)x_1(t) - \gamma x_3(t). \end{cases} \tag{6.3.1}$$

This is a variation of the Rössler system, which simplifies the Lorenz model of turbulence. When $\alpha_1 = 0.2$, $\alpha_2 = 1$, $\beta_1 = \beta_2 = 0.2$, $\gamma = 1.2$ and $\tau_1 = 1$, $\tau_2 = 2$, the system exhibits chaotic behaviour.

Data is generated from the constant initial function $\varphi(s) \equiv [1.5, 0.4, 0.9]^\intercal$. The resulting dataset is a single trajectory with a section of length $T_{\text{tr}} = 30$ used for training and a section of length $T_{\text{ts}} = 5$ used for testing. The data are sampled with $\Delta t = 0.05$ and central difference is used to calculate the state derivatives numerically.

6.3.1 SINDy for DDEs approach

We optimise both delays τ_1 and τ_2 and compare the results when using different degrees of polynomial libraries with $d = 2, 3, 4$. All of these libraries include the true second-degree nonlinearity that appears in the RHS of (6.3.1), but for $d = 3$ or $d = 4$ the library also contains many other terms that are missing from (6.3.1). When using E-SINDy, all delays are optimised simultaneously, and when using P-SINDy, only the maximum delay is optimised. For E-SINDy, candidate delays for τ_1 and τ_2 are uniformly sampled from the intervals $[0.1, 1.5]$ and $[1, 3]$. For P-SINDy, the candidate for the maximum delay $\bar{\tau}$ is uniformly sampled from the interval $[0.1, 2.5]$. PS optimisation is utilised to search for unknown delays. For P-SINDy, the degree of pseudospectral collocation for the discretisation from DDE to ODE is set to $M = 5$.

6.3.2 NDDEs approach

We consider the NDDE

$$\hat{x}'(t) = W_3 \tanh\Big(W_2 \tanh\big(W_1 \hat{\mathbf{x}}_t + b_1\big) + b_2\Big), \tag{6.3.2}$$

with different inputs and different number of neurons in the hidden layers. While training with derivative loss, the input vector is from data ($\mathbf{x}_t$ instead of $\hat{\mathbf{x}}_t$ in (6.3.2)) and the output vector is the predicted state derivative $\hat{x}'(t)$. Depending on whether we know that the delays only enter the state x_1, or consider that they may enter x_1, x_2 and x_3, we can construct different network inputs:

$$\mathbf{x}_t = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_1(t-\hat{\tau}_1) \\ x_1(t-\hat{\tau}_2) \end{bmatrix}, \qquad \mathbf{x}_t = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_1(t-\hat{\tau}_1) \\ x_2(t-\hat{\tau}_1) \\ x_3(t-\hat{\tau}_1) \\ x_1(t-\hat{\tau}_2) \\ x_2(t-\hat{\tau}_2) \\ x_3(t-\hat{\tau}_2) \end{bmatrix}. \tag{6.3.3}$$

We refer the first input as simplified input and the second as full input and we use both networks to learn from training data considering the derivative loss, as described in [74].

6.3.3 Comparison of the approaches

Interpretability and model structure

aspect	E-SINDy	P-SINDy	NDDE
direct RHS recovery (scalar)	✓	✓	×
direct RHS recovery (system)	✓	×	×
explicit delay identification	✓	✓	✓
sparse coefficient matrix	✓	✓	×
number of model parameters	small	small	very large
model interpretability	excellent	good	poor
functional form specification	required	required	not required

Table 6.3.1: comparison of model interpretability and structure across identification approaches.

E-SINDy provides the most interpretable results, directly recovering the RHS of the DDE as a sparse linear combination of basis functions. The sparse structure naturally highlights which terms are active in the dynamics.

P-SINDy sacrifices this level of interpretability when it comes to the case of systems. Although the collocation discretisation produces an ODE system and SINDy regression is applied to this first ODE, the infinite-dimensional structure is compressed into the augmented state space of collocation nodes. The resulting model can predict trajectories accurately, but does not directly reveal the functional form of the original DDE. This represents a pragmatic trade-off: one obtains a black-box model optimised for trajectory prediction rather than equation discovery.

Neural networks with trainable delays occupy a middle ground. The number of delay parameters are explicitly available, and one can visualise the trained network structure. However, the high-dimensional parameter space means that examining individual network weights provides limited insight into the system dynamics. Some interpretability emerges from the input-output relationships and the discovered delay values, but the nonlinear function mapping from delayed states to derivatives is generally opaque.

Delay parameter estimation

aspect	E-SINDy	P-SINDy	NDDE
known delays	direct application	direct application	training input
unknown delays	external opt.	max delay only	gradient-based
non-delay parameters	external opt.	external opt.	not required
CPU time (single delay)	moderate	low	high
CPU time (multiple delays)	moderate	low	high

Table 6.3.2: comparison of delay parameter estimation approaches.

The approaches employ fundamentally different strategies for handling unknown delays. E-SINDy requires external nested optimisation: for each candidate delay value (or tuple of delays), the full sparse regression problem must be solved. This creates a substantial computational burden when multiple delays are unknown, as the cost scales with the product of the search space dimensions.

P-SINDy reduces this burden significantly by recognising that only the maximum delay $\bar{\tau}$ requires external optimisation. The collocation grid automatically captures the history effects of intermediate delays through the evolution of the collocation state variables. This pragmatic observation reduces the external optimisation to a one-dimensional search (for $\bar{\tau}$) even when multiple delays are present in the original system.

NDDE training uses gradient-based optimisation (typically Adam algorithm [79]) to simultaneously learn delays and network weights. This avoids nested loops and leverages the automatic differentiation infrastructure of modern deep learning frameworks. However, the non-convex nature of neural network training introduces risks of convergence to local minima, particularly when training data is limited. Empirical results show that NDDE performance improves substantially when more diverse training data (multiple trajectories from different initial conditions) is available.

The trajectory recovery is shown in Figure 6.3.1 for the E-SINDy approach (with PS and polynomial libraries of orders 2 and 4) and for the NDDE approach (with simplified input, 5 neurons per layer and derivative loss). When the degree of the SINDy library is 2, it outperforms the NDDE in terms of prediction accuracy, while for degree 4 the SINDy predictions are less accurate than the NDDE predictions.

The E-SINDy approach with low order library can recover the data very well since the components at the RHS of (6.3.1) exist in the library and no higher-order nonlinearities are attempted to fit the data. However, when these higher-order terms are also included,

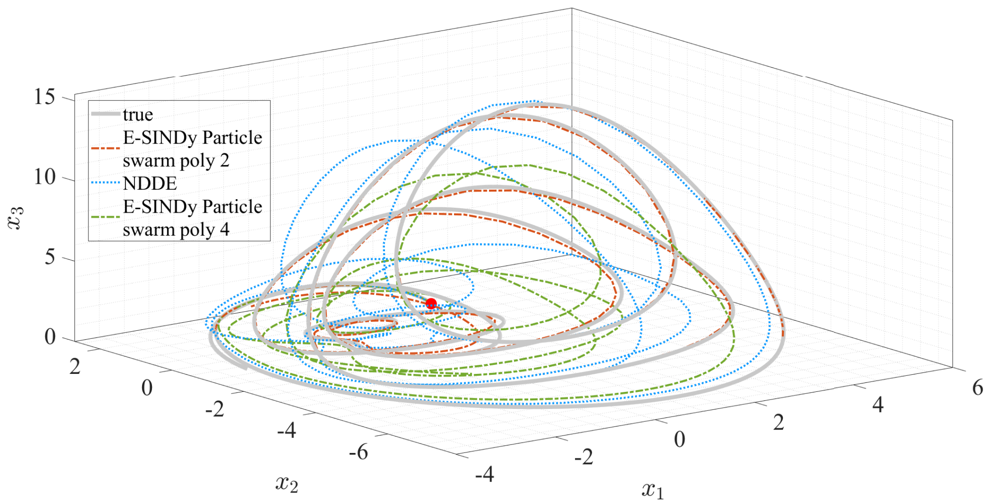

Figure 6.3.1: trajectory reconstruction for the Rössler model (6.3.1) showing the true trajectory (grey), E-SINDy trajectory when using library with polynomial degree 2 (red), E-SINDy trajectory when using library with polynomial degree 4 (green), and NDDE trajectory (blue). The true delay values are $(\tau_1, \tau_2) = (1, 2)$. The optimised delay values obtained through particle swarm optimisation for E-SINDy are $(\widehat{\tau}_1, \widehat{\tau}_2) = (1.0001, 2.0000)$ when using polynomial library of order 2 and $(\widehat{\tau}_1, \widehat{\tau}_2) = (1.0004, 2.0000)$ when using polynomial library of order 4. The NDDE approach yields $(\widehat{\tau}_1, \widehat{\tau}_2) = (0.2309, 1.9064)$ when using simplified input and 5 neurons in each hidden layer.

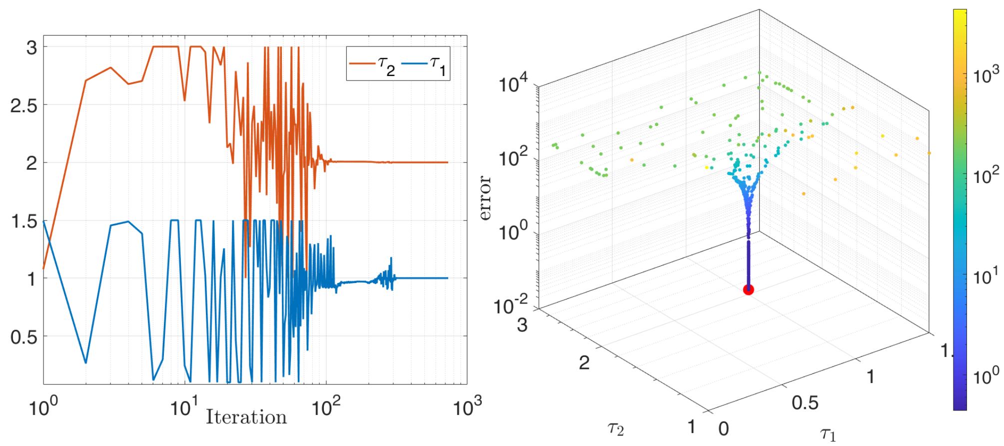

Figure 6.3.2: learned delays via particle swarm and particle filter errors along the iterations when applying E-SINDy with polynomial library of order 2 to the Rössler model (6.3.1).

the prediction accuracy decreases, as E-SINDy tries to capture the behaviour by blending polynomials of different degrees. In the meantime, the PS optimisation still accurately captures the time delays in both cases. For the limited amount of data utilised, the accuracy of NDDEs predictions fall between the two SINDy cases while the learnt delays

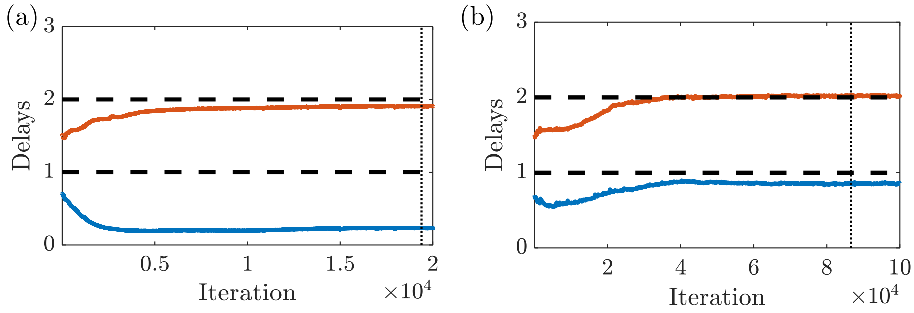

Figure 6.3.3: learned delays along the iterations for the two-delay case when applying NDDE (6.3.2) to the Rössler model (6.3.1). (a) Simplified input and 5 neurons in each hidden layer, trained on one trajectory. (b) Simplified input and 20 neurons in each hidden layer, trained on ten trajectories.

remain somewhat far from the true values. However, by including more training data, the performance of the NDDE can be significantly improved as will be demonstrated below.

The training procedure for E-SINDy with PS is shown in Figure 6.3.2. One may observe a clear minimum around the true delay values, which can be approached within a few hundred iterations with the help of the PS optimiser. For the NDDE, we depict the iterations in the trainable delays for different training datasets in Figure 6.3.3. While for the limited data described above only one of the two delays is captured accurately, see panel (a), both delays can be recovered when the amount of training data is increased 10 times (by using 10 different initial conditions to generate 10 distinct trajectories), as shown in panel (b).

Table 6.3.3 summarises the SINDy results for E and P5 with PS, including the absolute errors on both delays τ_1 and τ_2, as well as the reconstruction error $\text{RMSE}_{x'}$ for training and the errors $\text{RMSE}_{x'}$ and RMSE_x for testing. Different libraries of polynomial orders are compared in the table. Both methods achieve excellent reconstruction accuracy for polynomial order 2. E-SINDy shows superior computational efficiency but P-SINDy achieves slightly better delay estimation accuracy in most cases. When the order of polynomials increases in the library, the delay estimation remains accurate, while the reconstruction accuracy drops. This illustrates that having the right set of nonlinear functions in the library is important for SINDy to accurately identify the underlying dynamics of time delay systems.

The NDDE results are collected in Table 6.3.4. For each network input choice, we test two different network sizes, i.e., the number of neurons in each hidden layer is either 5 or 20, as indicated in the first column. The networks with simple input structure lead to smaller errors in learnt delays, that is, having restriction on the network input helps to regulate the network. In addition, increasing the number of neurons in each hidden layer from 5 to 20 yields better predictions on the testing data. Training the NDDE on a dataset consisting of 10 different trajectories results in more accurate learnt delays and simulation results compared to the NDDE trained on only one trajectory. That is, while learning from a single trajectory is challenging for neural networks (without any knowledge about

SINDy			Training	Testing		Calls	Time (s)
	$\lvert\tau_1 - \widehat{\tau}_1\rvert$	$\lvert\tau_2 - \widehat{\tau}_2\rvert$	$\mathrm{RMSE}_{x'}$	$\mathrm{RMSE}_{x'}$	RMSE_{x}		
Polynomial order 2							
E	6.73×10^{-3}	3.16×10^{-5}	4.31×10^{-3}	2.82×10^{-3}	8.17×10^{-2}	704	**86**
P5	-	$\mathbf{3.63 \times 10^{-7}}$	$\mathbf{5.33 \times 10^{-5}}$	$\mathbf{6.49 \times 10^{-5}}$	$\mathbf{5.84 \times 10^{-4}}$	**297**	156
Polynomial order 3							
E	5.89×10^{-4}	9.36×10^{-5}	1.55×10^{-2}	4.88×10^{-2}	$\mathbf{8.65 \times 10^{-1}}$	603	**191**
P5	-	$\mathbf{2.75 \times 10^{-7}}$	$\mathbf{4.84 \times 10^{-3}}$	$\mathbf{6.38 \times 10^{-3}}$	3.20×10^{-2}	**332**	304
Polynomial order 4							
E	7.32×10^{-4}	1.04×10^{-5}	$\mathbf{3.02 \times 10^{-1}}$	**2.64**	5.36	658	2143
P5	-	$\mathbf{6.42 \times 10^{-7}}$	7.66×10^{-1}	9.63	**3.28**	**247**	**568**

Table 6.3.3: performance of E-SINDy and P-SINDy with $M = 5$ and different polynomial orders for the Rössler system (6.3.1), spelling out the absolute error between the true and optimised delays, the RMSEs for training and testing phases, the number of SINDy calls and the CPU time while employing particle swarm.

NDDE			Training	Testing			
	$\lvert\tau_1 - \hat{\tau}_1\rvert$	$\lvert\tau_2 - \hat{\tau}_2\rvert$	$\mathcal{L}_{x'}$	RMSE$_{x'}$	RMSE$_x$	Iter.	Time (s)
Full (5)	8.70×10^{-1}	1.10	4.53×10^{-5}	$\mathbf{4.67 \times 10^{-2}}$	1.86×10^{-1}	20000	≈ 70
Full (20)	4.61×10^{-1}	6.32×10^{-1}	5.53×10^{-6}	7.09×10^{-1}	2.79		
Simplified (5)	7.69×10^{-1}	9.36×10^{-2}	1.91×10^{-4}	6.84×10^{-2}	$\mathbf{8.96 \times 10^{-2}}$		
Simplified (20)	3.24×10^{-1}	3.29×10^{-1}	$\mathbf{4.15 \times 10^{-6}}$	1.69×10^{-1}	1.87×10^{-1}		
Simplified* (20)	$\mathbf{1.39 \times 10^{-1}}$	$\mathbf{2.30 \times 10^{-2}}$	1.73×10^{-5}	2.21×10^{-1}	1.24×10^{-1}	100000	≈ 350

Table 6.3.4: performance of NDDE for the Rössler system (6.3.1), spelling out the absolute error between the true and optimised delays and the errors in training and testing phases. The last row (Simplified*) is trained on a larger dataset, where 10 trajectories are generated from 10 different constant histories.

the nonlinearities) [72, 73, 74], utilising data with multiple trajectories stemming from different initial histories improves the learning performance. In summary, having richer training data may compensate the missing a priori knowledge about the form of the RHS.

Although each SINDy step is "just" a sparse linear regression, the full E-SINDy and P-SINDy pipelines require many such regressions inside external optimisation loops over delays and parameters, repeatedly rebuilding and evaluating large libraries. By contrast, the NDDE approach trains a single end-to-end neural model where the loss is computed on derivatives and evaluated efficiently in batches, so all parameters are updated jointly in one optimisation loop. As a result, even though individual NDDE iterations are more expensive, the total time is lower than that of the multi-stage SINDy procedures.

6.4 Conclusions

This chapter has presented a comparative analysis of data-driven identification methodologies for DDEs, positioning the SINDy approaches developed in this thesis within the broader landscape of machine learning-based methods. The investigation establishes both the capabilities and limitations of different paradigms for discovering governing equations and delay parameters from time-series measurements.

The main contributions of this chapter are:

- unified framework for comparing sparse identification approaches (E-SINDy and P-SINDy) with NDDEs for DDE identification;
- detailed comparative analysis of interpretability, delay parameter estimation, computational efficiency, and data requirements across different methodologies;
- demonstration that E-SINDy achieves superior interpretability through direct recovery of governing equations as sparse linear combinations of basis functions, enabling immediate physical insight;
- validation that P-SINDy provides the most computationally efficient approach for systems with multiple unknown delays by requiring optimisation of only the maximum delay;
- showing that NDDE methods offer universal approximation capabilities without predefined of functional forms, but yielding substantially reduced interpretability and increased data requirements;
- comprehensive benchmarking on the delayed Rössler system demonstrating that appropriate library selection critically affects SINDy performance, while NDDE performance improves substantially with richer training data spanning multiple trajectories.

The comparative analysis reveals fundamental trade-offs between interpretability, computational efficiency, and data requirements. E-SINDy excels when interpretable governing equations are the primary objective and when reasonable prior knowledge about system structure is available. The sparse coefficient matrices directly reveal which physical mechanisms drive the dynamics, making E-SINDy particularly valuable for scientific discovery applications. However, computational costs scale unfavourably when multiple delays require simultaneous optimisation through external search procedures.

P-SINDy addresses this computational limitation through pseudospectral collocation, transforming the identification problem to require optimisation of only the maximum delay regardless of the number of intermediate delays present. This pragmatic reformulation achieves excellent trajectory reconstruction with substantially reduced computational burden. The trade-off manifests in reduced interpretability: the collocation discretisation mixes delay effects through the augmented state space, obscuring the explicit functional form of the original DDE. Nevertheless, P-SINDy represents an optimal choice when trajectory prediction rather than equation discovery constitutes the primary objective.

NDDE methods exploit the universal approximation property of neural networks to learn arbitrary nonlinear dynamics without predefined functional forms. The trainable delay parameters are discovered jointly with network weights via gradient-based optimisation, avoiding the complex loops required by sparse identification approaches. This flexibility is useful when system structure remains completely unknown. However, the high-dimensional parameter space fundamentally limits interpretability. Furthermore, neural networks typically require substantially larger training datasets, preferably spanning multiple trajectories from diverse initial conditions, to achieve reliable identification performance.

The Rössler system benchmarking demonstrates these trade-offs concretely. When the polynomial library contains the correct nonlinear terms (degree 2), E-SINDy achieves near-perfect delay identification and trajectory reconstruction with minimal computational cost. Including extraneous higher-order terms (degrees $3-4$) degrades performance, highlighting the importance of appropriate library construction. P-SINDy maintains excellent delay estimation accuracy across all library choices, demonstrating robustness to library specification. NDDE trained on a single trajectory achieves moderate performance, but accuracy improves dramatically when training data incorporates multiple trajectories from different initial conditions.

Computational efficiency varies substantially across methods. For the Rössler system with two delays, P-SINDy with PS requires $247-332$ SINDy calls completed in $156-568$ seconds depending on polynomial order. E-SINDy requires 603-704 calls completed in $86-2143$ seconds. NDDE training requires $20\,000-100\,000$ iterations completed in approximately $70-350$ seconds. While time appears comparable, the interpretability and data requirements differ fundamentally: sparse methods produce interpretable models from single trajectories, whereas neural networks require multiple trajectories to achieve comparable delay estimation accuracy.

6.4.1 Limitations and future directions

Several important limitations emerge from this comparative analysis:

(*i*) E-SINDy and P-SINDy performance depends critically on including appropriate nonlinear functions in the candidate library.

(*ii*) NDDEs require substantially more training data to achieve reliable identification.

(*iii*) all methods shown here assume the number of delays is specified or bounded.

(*iv*) complementary strengths of sparse and neural network methods suggest potential benefits from hybrid approaches combining sparse regression with neural network components or gradient-based delay learning with sparse coefficient estimation.

(*v*) comparative analysis employed relatively clean synthetic data.

These limitations indicate several key research directions. Fusing gradient-based delay learning from neural network approaches with sparse regression frameworks could simultaneously achieve computational efficiency and interpretability.

The choice of identification methodology should be guided by the specific application requirements. When interpretability and equation discovery constitute primary objectives, and when reasonable prior knowledge about system structure is available, E-SINDy provides the most direct approach to recovering governing equations. When computational efficiency matters more than complete interpretability, particularly for systems with multiple unknown delays, P-SINDy offers the most pragmatic solution. When system structure remains completely unknown and sufficient diverse training data is available, NDDE methods provide flexible black-box identification capabilities.

Together, the sparse identification frameworks developed in this thesis and the neural network approaches reviewed in this chapter provide a comprehensive toolkit for data-driven identification of delayed dynamical systems.

This chapter completes the methodological development for deterministic delay systems. The comprehensive comparison of identification paradigms establishes that multiple complementary approaches exist, each with distinct advantages matched to different application contexts. The next chapter synthesises the key findings across the entire thesis and discusses broader implications for data-driven identification of delayed dynamical systems.

7 Conclusions

This thesis presents a comprehensive investigation into data-driven identification of delay differential equations (DDEs), stochastic delay differential equations (SDDEs), and distributed delay systems (DIDEs and REs) through extensions of the Sparse Identification of Nonlinear Dynamics (SINDy) framework. The work addresses a critical gap in the intersection of system identification, data-driven discovery, and infinite-dimensional dynamical systems by developing robust, interpretable, and computationally efficient methods for discovering governing equations directly from observational time-series data. This conclusion is primarily based on the work published and submitted in [16, 18, 28, 29, 42] and the framework discussed in the book chapter [25].

The work begins with the recognition that many real-world systems exhibit memory effects through discrete delays, stochastic delays, or distributed delays. Traditional system identification approaches [92, 106] require substantial prior knowledge of system structure and delay parameters, which is rarely available in practical applications. Furthermore, existing data-driven methods struggle with the curse of dimensionality introduced by multiple unknown delays and the infinite-dimensional nature of delay systems. This work resolves these limitations through systematic methodological extensions to the SINDy framework, enabling automatic discovery of sparse governing equations from data.

7.1 Principal contributions

The work establishes the following key contributions:

7.1.1 Expert SINDy with unknown delay identification

The first major contribution develops E-SINDy with external optimisation for unknown delays. This method combines standard SINDy regression with particle swarm optimisation (PS) to discover delay parameters without requiring prior knowledge. The work demonstrates that PS significantly outperforms brute force (BF) and Bayesian optimisation (BO) approaches in terms of both computational efficiency and robustness. The method achieves good accuracy of parameter recovery when the library accurately represents the true dynamics, and the sparse coefficient matrix directly reveals interpretable governing mechanisms. However, the sensitivity to library specification remains the primary limitation, missing nonlinear terms lead to spurious fitted terms, while extraneous terms degrade accuracy.

7.1.2 Pragmatic SINDy via pseudospectral collocation

The second major contribution introduces P-SINDy, a novel reformulation of DDE identification via pseudospectral collocation. This transforms the infinite-dimensional DDE into a finite-dimensional ODE system, reducing computational complexity by requiring optimisation of only the maximum delay. For systems with multiple unknown delays, P-SINDy achieves good computational efficiency compared to E-SINDy with multivariate delay space search. The trade-off is reduced interpretability of the reconstructed right-hand side, as collocation discretisation obscures the explicit delay structure, making P-SINDy more suitable for trajectory prediction than equation discovery.

7.1.3 Extensions to stochastic delay differential equations

The third contribution extends the SINDy framework to SDDEs, addressing the simultaneous identification of drift and diffusion terms from noisy data. The methodology employs Kramers-Moyal moment estimators with higher-order finite difference schemes to estimate drift and diffusion. Three complementary estimation strategies are developed: Approach A generates synthetic trajectories from a single observed path; Approach B1 averages increment statistics before sparse regression, providing robust model discovery; Approach B2 applies sparse regression separately to each trajectory and averages results. The work demonstrates that drift identification achieves excellent performance with relatively few samples, whereas diffusion term identification demands substantially more data due to amplification of measurement noise through moment ratios. The extended SINDy-SDDE framework provides good interpretability for drift identification but remains computationally demanding for diffusion, particularly in high dimensions.

7.1.4 Distributed delay kernel identification

The fourth major contribution extends SINDy to identifying delay integro-differential equations (DIDEs) and renewal equations (REs) with distributed memory. This represents a significant methodological innovation, as distributed delay identification from data remains largely unexplored. The framework uses quadrature-based kernel libraries and integral SINDy to recover nonautonomous kernel functions directly from trajectories. The method successfully discovers both the analytical structure of kernels and unknown integration bounds through joint external optimisation and internal sparse regression. Unlike standard SINDy which produces high-dimensional black-box models, the specialised distributed delay framework (DD-SINDy) yields interpretable sparse representations revealing the underlying kernel structure and biological mechanisms, as demonstrated through comparison with black-box SINDy (BB-SINDy) on Ricker-type population models.

7.1.5 Practical applications to epidemiological data

The fifth contribution demonstrates application of the distributed delay SINDy framework to real-world epidemiological data: Severe Fever with Thrombocytopenia Syndrome (SFTS) transmission dynamics in Dalian, China. This case study illustrates how data-driven methods can discover simplified, yet predictive representations of complex structured population dynamics without requiring detailed a priori knowledge of transmission mechanisms. The identified models capture distributed delays modulated by temperature reflecting the structure and life stages of the tick population, demonstrating the practical utility of theoretical developments to predict disease transmission and guide public health interventions.

7.1.6 Comparative framework and method selection guidance

The work provides a comparative analysis between E-SINDy, P-SINDy, and neural delay differential equations (NDDEs). This positioning within the broader landscape of data-driven system identification enables practitioners to select appropriate methods based on problem characteristics. E-SINDy excels when equation discovery and interpretability are priorities, delays are known or semi-known, and prior knowledge of nonlinear terms is available. P-SINDy optimises computational efficiency for systems with multiple unknown delays where trajectory prediction suffices. NDDEs approaches are appropriate when unknown nonlinearities cannot be represented through polynomial basis, abundant measurement data are available, and multiple independent trajectories support training. The work demonstrates that no universal method exists; success requires careful problem formulation and close engagement with domain knowledge.

7.2 Limitations and challenges

This section discusses the principal limitations and challenges encountered throughout this work across all developed methodologies. Understanding these limitations is essential for practitioners to apply the methods appropriately and for future researchers to identify productive directions for methodological advancement.

E-SINDy and P-SINDy require *a priori* specification of candidate basis functions and cannot discover functional forms absent from the library, representing a fundamental limitation distinguishing sparse identification from black-box approaches. E-SINDy's requirement to simultaneously optimise multiple unknown delay parameters creates computational challenges that grow linearly with the number of delays. The dimensionality of the polynomial library grows combinatorially with the state dimension and the polynomial degree, restricting its applicability to systems. Numerical differentiation amplifies measurement noise, particularly at high polynomial orders, and multiple sparse solutions may fit available data equally well, creating model non-uniqueness when information is insufficient.

A critical limitation of SINDy-SDDE is the fundamental asymmetry in identification performance: drift terms achieve excellent results with moderate data quantities, while diffusion term identification demands orders of magnitude more data due to amplification of measurement noise through moment ratio computations. High-dimensional stochastic systems become computationally demanding and statistically unreliable, requiring enormous data volumes for acceptable performance. Effective identification requires multiple independent trajectories or very long trajectories split into independent segments, which presents practical challenges for many experimental systems. Approach B1 (pre-regression averaging), though superior empirically, requires careful tuning of neighbourhood size that balances bias and variance, with limited guidance for automated or adaptive selection strategies.

Identifying nonautonomous kernel functions from time-series data presents a fundamentally different challenge from discrete delay identification, since kernels represent continuous functions over delay values rather than finite-dimensional parameters. Quadrature errors accumulate through the identification process, yet the relationship between quadrature accuracy and final reconstruction error lacks formal theoretical characterisation. When integration bounds are unknown, they must be estimated through external optimisation, expanding the search space and creating challenges in convergence. Kernel library construction requires practitioners to hypothesise functional forms of distributed delay structures, which may be unknown or nonlinear, demanding more domain-specific insight than standard polynomial or trigonometric basis.

Real-world application to the SFTS data reveals that epidemiological measurements suffer from multiple sources of uncertainty sources including reporting delays, underreporting, and partial observation of the state of the system, preventing a formal distinction between the uncertainty of the structural model and the uncertainty of the measurement. Monthly surveillance data from 2011 to 2022 provide limited data, coarse relative to tick population dynamics operating on multi-week timescales, reducing ability to identify rapidly varying kernel structures. Multiple plausible models may fit observed case data equally well given data limitations, meaning identified structures may reflect mathematical form choices rather than unique biological features.

7.3 Conclusions and future perspectives

This work addresses data-driven identification of DDEs and distributed delay systems through systematic SINDy extensions, establishing that sparse identification excels when interpretability and scientific discovery constitute primary objectives, while neural approaches provide universal approximation without library construction at substantially higher data costs.

This work equips researchers with theoretical understanding and practical tools to advance data-driven system identification into new problem domains where memory effects, stochasticity, and complexity challenge traditional approaches, demonstrating that interpretable identification of infinite-dimensional dynamical systems is achievable and valuable for scientific discovery, prediction, and control applications across diverse scientific and engineering domains.

A

Appendix: Methodological pseudocodes

This appendix compiles the comprehensive pseudocode for the numerical methods and identification algorithms that have been developed and examined in this thesis. Its purpose is to offer an implementation-focused description of each SINDy variant in a standardised format, supplementing the theoretical discussions and numerical findings detailed in the main chapters. The algorithms are categorised based on the type of dynamical system: delay differential equations (DDEs), stochastic delay differential equations (SDDEs), and delay integro-differential equations (DIDEs) and renewal equations (REs). For each category, specific SINDy procedures are outlined, including methods for both known and unknown delay parameters.

MATLAB implementations for the methods presented in this thesis can be found at http://cdlab.uniud.it/software.

A.1 E-SINDy for DDEs

Algorithm A.1 E-SINDy: Expert SINDy for known discrete delays

Input: Data $X \in \mathbb{R}^{m \times n}$, derivative data $X' \in \mathbb{R}^{m \times n}$, known delays $\boldsymbol{\tau} = [\tau_1, \dots, \tau_k]^T$, parameters $\boldsymbol{\rho}$, library $\mathcal{F}$, threshold λ, method $\mathcal{M}$ (STLS/LASSO).

Output: Sparse coefficient matrix $\Xi(\boldsymbol{\tau}) \in \mathbb{R}^{p \times n}$, reconstruction errors.

Step 1: Delayed data generation

for $i = 1$ to k **do**

▷ piecewise linear interpolation for shifted data

  $X_{\tau_i} = \text{interp1}(t, X, t - \tau_i)$

end for

Step 2: Augmented library construction

$X_{\boldsymbol{\tau}} = [X_{\tau_1}, X_{\tau_2}, \dots, X_{\tau_k}]$ ▷ concatenate all delayed states

$\Theta(X, X_{\boldsymbol{\tau}}) = \text{LibraryApply}(\mathcal{F}, X, X_{\boldsymbol{\tau}}, \boldsymbol{\rho})$

Step 3: Column-wise sparse regression

for $j = 1$ to n **do do** ▷ for each state variable

  if $\mathcal{M}$ == STLS **then**

    $\xi_j = \text{STLS}(X'_j, \Theta, \lambda)$

  else (LASSO)

    $\xi_j = \arg\min_{\xi} \left(\|X'_j - \Theta\xi\|_2 + \lambda\|\xi\|_1 \right)$

  end if

end for

Step 4: Assembly and validation

$\Xi(\boldsymbol{\tau}) = [\xi_1, \xi_2, \dots, \xi_n]$

$\text{RMSE}_{x'} = \|X' - \Theta(X, X_{\boldsymbol{\tau}})\Xi(\boldsymbol{\tau})\|_2$

▷ simulate DDE using reconstructed model for validation

$X_{\text{sim}} = \mathcal{S}_{\text{DDE}}(\Xi(\boldsymbol{\tau}), \boldsymbol{\tau}, \boldsymbol{\rho})$

$\text{RMSE}_x = \|X - X_{\text{sim}}\|_2$

return $\Xi(\boldsymbol{\tau})$, $\text{RMSE}_{x'}$, RMSE_x

A.2 P-SINDy for DDEs

Algorithm A.2 P-SINDy: Pragmatic SINDy via pseudospectral collocation

Input: Data $X \in \mathbb{R}^{m \times n}$, derivative data X', library $\mathcal{F}$, parameters $\boldsymbol{\rho}$, collocation degree M, delay bounds $[\tau_{\min}, \tau_{\max}]$, threshold λ.

Output: Optimal maximum delay $\bar{\tau}^*$, sparse coefficients $\Xi_M(\bar{\tau}^*)$, ODE approximation.

function P_SINDY_OBJECTIVE($\bar{\tau}$)

  1. Grid generation (Chebyshev)

  $s_i = \frac{\bar{\tau}}{2}\left(\cos\left(\frac{i\pi}{M}\right) - 1\right)$ for $i = 0, \dots, M$

  2. Data collocation

  $X_s = [X, X_{s_1}, \dots, X_{s_M}]$

  3. Library & regression (first block)

  $\Theta(X, X_s) = \text{LibraryApply}(\mathcal{F}, X_s, \boldsymbol{\rho})$

  for $j = 1$ to n **do**

    $\xi_j = \text{SparseRegression}(X'_j, \Theta(X, X_s), \lambda)$

  end for

  $\Xi_M(\bar{\tau}) = [\xi_1, \dots, \xi_n]$

  return $\|X' - \Theta(X, X_s)\Xi_M(\bar{\tau})\|_2$ ▷ reconstruction error

end function

external optimisation

$\bar{\tau}^* = \arg\min_{\bar{\tau} \in [\tau_{\min}, \tau_{\max}]} \text{P\_SINDy\_Objective}(\bar{\tau})$

▷ solved via particle swarm, bayesian optimisation, etc.

Final model construction

$\Xi_M(\bar{\tau}^*)$ = retrieve coefficients using $\bar{\tau}^*$

construct ODE system:

  $U'_0(t) = \Theta(U(t), \boldsymbol{\rho})\Xi_M(\bar{\tau}^*)$ ▷ learned dynamics (f)

  $U'_i(t) = D_M U(t)$ for $i = 1, \dots, M$ ▷ differentiation matrix

return $\bar{\tau}^*$, $\Xi_M(\bar{\tau}^*)$, ODE system

A.3 SINDy for DDEs parameter optimisation

Algorithm A.3 Unified external optimisation for delays and parameters (E-SINDy & P-SINDy)

Input: Data $X \in \mathbb{R}^{m\times n}$, derivative data X', method $\mathcal{M} \in \{\text{E-SINDy, P-SINDy}\}$, parameter bounds $[\boldsymbol{\theta}_{\min}, \boolsymbol{\theta}_{\max}]$, optimiser $\mathcal{O}$.
Output: Optimal parameters $\boldsymbol{\theta}^*$ (delays $\boldsymbol{\tau}^*$, parameters $\boldsymbol{\rho}^*$), sparse coefficients Ξ^*.
function SINDY_OBJECTIVE($\boldsymbol{\theta}$)
  parse $\boldsymbol{\theta}$ into delays $\boldsymbol{\tau}$ and parameters $\boldsymbol{\rho}$
  if $\mathcal{M}$ == "E-SINDy" **then**
    $\Theta = \text{ConstructLibrary}(X, \boldsymbol{\tau}, \boldsymbol{\rho})$ ▷ Using shifted data $X_{\boldsymbol{\tau}}$
  else **(P-SINDy)**
    $\bar{\tau} = \max(\boldsymbol{\tau})$ ▷ only max delay required
    $\Theta = \text{PseudospectralLibrary}(X, \bar{\tau}, \boldsymbol{\rho})$ ▷ using collocation on $[-\bar{\tau}, 0]$
  end if
  $\Xi = \text{SparseRegression}(X', \Theta)$ ▷ e.g., STLS
  return $\|X' - \Theta\Xi\|_2$ ▷ reconstruction error
end function
external optimisation loop
define search space Ω based on bounds $[\boldsymbol{\theta}_{\min}, \boldsymbol{\theta}_{\max}]$
if $\mathcal{O}$ == "BruteForce" **then**
  generate grid $\subset \Omega$
  $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \text{SINDy\_Objective}(\boldsymbol{\theta})$
else if $\mathcal{O}$ == "BayesianOptimisation" **then**
  model $f(\boldsymbol{\theta}) \approx \text{SINDy\_Objective}(\boldsymbol{\theta})$
  while not converged **do**
    $\boldsymbol{\theta}_{\text{next}} = \arg\max \text{acquisition}(f, \Omega)$
    $y_{\text{next}} = \text{SINDy\_Objective}(\boldsymbol{\theta}_{\text{next}})$
    update model f with $(\boldsymbol{\theta}_{\text{next}}, y_{\text{next}})$
  end while
  $\boldsymbol{\theta}^* =$ best evaluated $\boldsymbol{\theta}$
else if $\mathcal{O}$ == "ParticleSwarm" **then**
  initialise population P in Ω
  while not converged **do**
    for particle $p \in P$ **do**
      fitness = SINDy_Objective(p.position)
      update p.velocity and p.position based on global/local best
    end for
  end while
  $\boldsymbol{\theta}^* =$ GlobalBest.position
end if
final model constructed
parse $\boldsymbol{\theta}^*$ into optimal $\boldsymbol{\tau}^*$ and $\boldsymbol{\rho}^*$
Run SINDy_Objective($\boldsymbol{\theta}^*$) to retrieve Ξ^*
return $\boldsymbol{\tau}^*, \boldsymbol{\rho}^*, \Xi^*$

A.4 SINDy for SDDEs: Approach A

Algorithm A.4 SINDy for SDDEs - Approach A: synthetic realisations from single trajectory

Input: Single trajectory $X \in \mathbb{R}^{m \times n}$, time step Δt, delay τ, synthetic ensemble size M, initial models f_0, g_0, libraries $\mathcal{F}_f, \mathcal{F}_G$, thresholds λ_f, λ_G.
Output: Drift coefficients Ξ_f, Diffusion coefficients Ξ_G.
Step 1: State augmentation
$X_\tau = \text{interp1}(t, X, t - \tau)$ ▷ construct delayed state samples
$Z = [X, X_\tau]$ ▷ augmented state $Z(t_i) \in \mathbb{R}^{2n}$
Step 2: Synthetic generation & moment estimation
for $i = 1, \dots, m-1$ **do**
  generate M independent brownian increments $\{\Delta W_i^{(k)}\}_{k=1}^{M}$
  for $k = 1, \dots, M$ **do**
    initialise $X^{(k)}(t_i) = X(t_i)$, $X^{(k)}(t_i - \tau) = X(t_i - \tau)$
    generate next step using Euler-Maruyama:
      $X^{(k)}(t_{i+1}) = X^{(k)}(t_i) + f_0(Z^{(k)}(t_i))\Delta t + g_0(Z^{(k)}(t_i))\Delta W_i^{(k)}$
    record increment: $\Delta X_i^{(k)} = X^{(k)}(t_{i+1}) - X^{(k)}(t_i)$
  end for
  estimate drift and diffusion using specified estimator:
  $\mathbf{f}(Z(t_i)) = \frac{1}{M\Delta t} \sum_{k=1}^{M} \Delta X_i^{(k)}$
  $\mathbf{G}(Z(t_i)) = \frac{1}{M\Delta t} \left[\sum_{k=1}^{M} \Delta X_i^{(k)} (\Delta X_i^{(k)})^T - M\mathbf{f}(Z(t_i))\mathbf{f}(Z(t_i))^T \Delta t \right]$
end for
construct library matrices $\Theta_f(Z), \Theta_G(Z)$ from functions in $\mathcal{F}_f, \mathcal{F}_G$
▷ sparse regression for drift:
for $i = 1, \dots, n$ **do**
  $\Xi_f = \arg\min_\alpha \left(\|\mathbf{f}_i(Z) - \Theta_f(Z)\alpha\|_2 + \lambda_f \|\alpha\|_1 \right)$
end for
▷ sparse regression for diffusion:
for $i, j = 1, \dots, n$ **do**
  $\Xi_G = \arg\min_\beta \left(\|\mathbf{G}_{i,j}(Z) - \Theta_G(Z)\beta\|_2 + \lambda_G \|\beta\|_1 \right)$
end for
return Ξ_f, Ξ_G

A.5 SINDy for SDDEs: Approach B1

Algorithm A.5 SINDy for SDDEs - Approach B1: pre-regression averaging

Input: Ensemble of trajectories $\{X^{(k)}\}_{k=1}^{M} \in \mathbb{R}^{m\times n}$, delay τ, time step Δt, tolerance ε, estimator $\mathcal{E} \in \{\text{KM, FD, CD, TR}\}$, libraries $\mathcal{F}_f, \mathcal{F}_G$, thresholds λ_f, λ_G.
Output: Drift coefficients Ξ_f, diffusion coefficients Ξ_G.
Step 1: State augmentation (all trajectories)
for $k = 1$ to M **do**
  $X_\tau^{(k)} = \text{interp1}(t, X^{(k)}, t-\tau)$
  $Z^{(k)} = [X^{(k)}, X_\tau^{(k)}]$ ▷ augmented state for trajectory k
end for
Step 2: Neighbourhood estimation (pre-regression averaging)
for each evaluation point z in dataset $\{Z^{(k)}(t_i)\}$ **do**
  1. find neighbours:
  $S_\varepsilon(z) = \{(k,i) : \|Z^{(k)}(t_i) - z\| \le \varepsilon\}$ $N_\varepsilon(z) = |S_\varepsilon(z)|$
  2. compute moments based on estimator $\mathcal{E}$:
  if $\mathcal{E} == \text{KM}$ **then** ▷ Kramers-Moyal
    $\Delta X_i^{(k)} = X^{(k)}(t_{i+1}) - X^{(k)}(t_i)$
    $\hat{f}(z) = \frac{1}{N_\varepsilon \Delta t}\sum_{(k,i)\in S_\varepsilon} \Delta X_i^{(k)}$
    $\hat{G}(z) = \frac{1}{2N_\varepsilon \Delta t}\sum_{(k,i)\in S_\varepsilon} \Delta X_i^{(k)} (\Delta X_i^{(k)})^T$
  else if $\mathcal{E} == \text{FD}$ **then** ▷ second-order forward difference
    $\Delta X_i^{(k)} = X^{(k)}(t_{i+1}) - X^{(k)}(t_i)$
    $\Delta X_{2i}^{(k)} = X^{(k)}(t_{i+2}) - X^{(k)}(t_i)$
    $\hat{f}(z) = \frac{1}{N_\varepsilon}\sum_{(k,i)\in S_\varepsilon} \frac{4\Delta X_i^{(k)} - \Delta X_{2i}^{(k)}}{2\Delta t}$
    $\hat{G}(z) = \frac{1}{4N_\varepsilon \Delta t}\sum_{(k,i)\in S_\varepsilon} [4\Delta X_i^{(k)} (\Delta X_i^{(k)})^T - \Delta X_{2i}^{(k)} (\Delta X_{2i}^{(k)})^T]$
  else if $\mathcal{E} == \text{CD}$ **then** ▷ central difference
    $\Delta X_{+i}^{(k)} = X^{(k)}(t_{i+1}) - X^{(k)}(t_{i-1})$
    $\hat{f}(z) = \frac{1}{N_\varepsilon}\sum_{(k,i)\in S_\varepsilon} \frac{\Delta X_{+i}^{(k)}}{2\Delta t}$
    $\hat{G}(z) = \frac{1}{4N_\varepsilon \Delta t}\sum_{(k,i)\in S_\varepsilon} \Delta X_{+i}^{(k)} (\Delta X_{+i}^{(k)})^T$
  else if $\mathcal{E} == \text{TR}$ **then** ▷ Trapezoidal rule
    apply trapezoidal rule on neighbourhood S_ε
  end if
  store estimates: $\hat{f}(z)$, $\hat{G}(z)$
end for
construct library matrices $\Theta_f(Z), \Theta_G(Z)$ from functions in $\mathcal{F}_f, \mathcal{F}_G$
for $i = 1, \ldots, n$ **do**
  $\Xi_f = \arg\min_\alpha \left(\|\hat{f}_i(Z) - \Theta_f(Z)\Xi_f\|_2 + \lambda_f \|\Xi_f\|_1\right)$ ▷ estimate drift
end for
for $i, j = 1, \ldots, n$ **do**
  $\Xi_G = \arg\min_\beta \left(\|\hat{G}_{i,j}(Z) - \Theta_G(Z)\Xi_G\|_2 + \lambda_G \|\Xi_G\|_1\right)$ ▷ estimate diffusion
end for
return Ξ_f, Ξ_G

A.6 SINDy for SDDEs: Approach B2

Algorithm A.6 SINDy for SDDEs - Approach B2: post-regression averaging

Input: Ensemble of trajectories $\{X^{(k)}\}_{k=1}^{M} \in \mathbb{R}^{m\times n}$, delay τ, time step Δt, estimator $\mathcal{E} \in \{\text{KM, FD, CD, TR}\}$, libraries $\mathcal{F}_f, \mathcal{F}_G$, thresholds λ_f, λ_G.
Output: averaged drift coefficients $\bar{\Xi}_f$, averaged diffusion coefficients $\bar{\Xi}_G$.
Step 1: Individual trajectory processing
initialize coefficient storage $\mathcal{A} = \emptyset$, $\mathcal{B} = \emptyset$
for $k = 1$ to M **do**
    1. state augmentation
    $X_\tau^{(k)} = \text{interp1}(t, X^{(k)}, t - \tau)$
    $Z^{(k)} = [X^{(k)}, X_\tau^{(k)}]$ ▷ augmented state for trajectory k
    2. trajectory-wise estimation
    if $\mathcal{E} == \text{KM}$ **then**
        $\hat{f}^{(k)} = \text{KramersMoyalDrift}(X^{(k)}, \Delta t)$
        $\hat{G}^{(k)} = \text{KramersMoyalDiff}(X^{(k)}, \Delta t)$
    else if $\mathcal{E} == \text{FD}$ **then**
        $\hat{f}^{(k)}, \hat{G}^{(k)} = \text{ForwardDiffEstimator}(X^{(k)}, \Delta t)$
    else if $\mathcal{E} == \text{CD}$ **then**
        $\hat{f}^{(k)}, \hat{G}^{(k)} = \text{CentralDiffEstimator}(X^{(k)}, \Delta t)$
    end if ▷ see Section 3.2 for estimator formulas
    construct library matrices $\Theta_f(Z^{(k)}), \Theta_G(Z^{(k)})$ from functions in $\mathcal{F}_f, \mathcal{F}_G$
    sparse regression for drift (trajectory k):
    for $i = 1, \ldots, n$ **do**
        $\Xi_f^{(k)} = \arg\min_{\Xi_f} \left(\|\hat{f}_i^{(k)}(Z^{(k)}) - \Theta_f(Z^{(k)})\Xi_f\|_2 + \lambda_f \|\Xi_f\|_1 \right)$
    end for
    sparse regression for diffusion (trajectory k):
    for $i, j = 1, \ldots, n$ **do**
        $\Xi_G^{(k)} = \arg\min_{\Xi_G} \left(\|\hat{G}_{i,j}^{(k)}(Z^{(k)}) - \Theta_G(Z^{(k)})\Xi_G\|_2 + \lambda_G \|\Xi_G\|_1 \right)$
    end for
end for
Step 2: Ensemble averaging
$\bar{\Xi}_f = \frac{1}{M}\sum_{k=1}^{M} \Xi_f^{(k)}$ ▷ Averaging in parameter space
$\bar{\Xi}_G = \frac{1}{M}\sum_{k=1}^{M} \Xi_G^{(k)}$
return $\bar{\Xi}_f, \bar{\Xi}_G$

A.7 Drift and diffusion estimation methods for SDDEs

Algorithm A.7 Estimation methods for drift and diffusion

```
Input: Data X(t_i) ∈ ℝ^n, augmented state Z(t_i), time step Δt, estimator ℰ.
Output: Drift estimates f̂(Z(t)), diffusion estimates Ĝ(Z(t)).
function APPLYESTIMATOR(X, Δt, ℰ)
    if ℰ == KM then                                              ▷ Kramers-Moyal
        for i = 1, ..., m − 1 do
            f̂(Z(t_i)) = (X(t_{i+1}) − X(t_i)) / Δt
            Ĝ(Z(t_i)) = (X(t_{i+1}) − X(t_i))(X(t_{i+1}) − X(t_i))^T / (2Δt)
        end for
    else if ℰ == FD then                          ▷ second-order forward difference
        for i = 1, ..., m − 2 do
            f̂(Z(t_i)) = [4(X(t_{i+1}) − X(t_i)) − (X(t_{i+2}) − X(t_i))] / (2Δt)
            Ĝ(Z(t_i)) = 1/(4Δt) [4(X(t_{i+1}) − X(t_i))(X(t_{i+1}) − X(t_i))^T
                        −(X(t_{i+2}) − X(t_i))(X(t_{i+2}) − X(t_i))^T]
        end for
    else if ℰ == CD then                                      ▷ central difference
        for i = 2, ..., m − 1 do
            f̂(Z(t_i)) = (X(t_{i+1}) − X(t_{i−1})) / (2Δt)
            Ĝ(Z(t_i)) = (X(t_{i+1}) − X(t_{i−1}))(X(t_{i+1}) − X(t_{i−1}))^T / (4Δt)
        end for
    else if ℰ == TR then                                         ▷ trapezoidal rule
        initialise f̂^(0)(Z) = KM estimates
        repeat
            for i = 1, ..., m − 1 do
                f̂^(k+1)(Z(t_i)) = 2 (X(t_{i+1}) − X(t_i)) / Δt − f̂^(k)(Z(t_{i+1}))
                update Ĝ using trapezoidal rule with current f̂ estimates
            end for
        until convergence (‖f̂^(k) − f̂^(k−1)‖ < tol)
    end if
    return f̂, Ĝ
end function
```

A.8 SINDy for distributed delay

Algorithm A.8 SINDy for distributed delay and renewal equations

Input: Data $\mathbf{X} \in \mathbb{R}^{m \times n}$, derivative data $\mathbf{X}' \in \mathbb{R}^{m \times n}$ (for DDEs), function library $\mathcal{F}$, delay interval $[-\tau, 0]$, quadrature rule parameters, regularisation parameter λ
Output: Sparse coefficient matrix Ξ representing kernel $\gamma(\sigma, x(t+\sigma))$
Step 1: Quadrature setup
select quadrature rule (trapezoidal, Clenshaw-Curtis, etc.)
generate K quadrature nodes $\{\sigma_k\}_{k=1}^{K}$ in $[-\tau, 0]$
compute corresponding quadrature weights $\{w_k\}_{k=1}^{K}$
Step 2: Data preparation
for $k = 1$ to K **do**
  obtain time shifted data $\mathbf{X}_{\sigma_k}$ by interpolating $\mathbf{X}$ at times $t_i + \sigma_k$
end for
Step 3: Library construction
for $k = 1$ to K **do**
  construct local library $\Theta(\sigma_k, \mathbf{X}_{\sigma_k})$ using functions from $\mathcal{F}$:
    include terms like: $1, \sigma_k, X_1(\cdot + \sigma_k), \ldots, X_n(\cdot + \sigma_k)$,
    $\sigma_k^2, \sigma_k X_1(\cdot + \sigma_k), \ldots, \sigma_k X_n(\cdot + \sigma_k), X_1^2(\cdot + \sigma_k), \ldots$
    $\sigma_k^d, \sigma_k^{d-1} X_1(\cdot + \sigma_k), \ldots, X_n^d(\cdot + \sigma_k)$
  weight the library: $\Theta_k = w_k \cdot \Theta(\sigma_k, \mathbf{X}_{\sigma_k})$
end for
Combine libraries: $\Theta = \sum_{k=1}^{K} w_k \Theta_k$
Step 4: Sparse regression
for $j = 1$ to n (each state component) **do**
  if DDE case **then**
    set target vector: $Y_j = \mathbf{X}'_j$
  else(RE case)
    set target vector: $Y_j = \mathbf{X}_j$
  end if
  solve sparse regression using STLS or similar:
    $\xi_j = \arg\min_{\xi} \left(\left\| Y_j - \left(\sum_{k=1}^{K} w_k \mathbf{\Theta}_k \right) \xi \right\|_2 + \lambda \|\xi\|_1 \right)$
end for
assemble coefficient matrix: $\Xi := (\xi_1, \xi_2, \ldots, \xi_n)$
Step 5: Model validation
simulate identified model using numerical integration
compute prediction errors on training and validation data
return Ξ, validation metrics

A.9 SINDy for distributed delay parameter optimisation

Algorithm A.9 SINDy with parameter optimisation for distributed delays

Input: Data X, derivative data $\mathbf{X}'$, function library $\mathcal{F}$, parameter search spaces
Output: Optimal parameters ρ^* and sparse coefficient matrix Ξ^*
function DISTRIBUTED_SINDY_ERROR(ρ)
    parse parameters: delay bounds $[\tau_1, \tau_2]$, quadrature settings from ρ
    generate quadrature nodes $\{\sigma_k\}_{k=1}^{K}$ in $[\tau_1, \tau_2]$
    compute quadrature weights $\{w_k\}_{k=1}^{K}$
    obtain time shifted data $\{\mathbf{X}_{\sigma_k}\}_{k=1}^{K}$ by interpolation
    construct weighted library matrix Θ as in Algorithm 1
    for $j = 1$ to n **do**
        solve sparse regression to get $\xi_j(\rho)$
    end for
    simulate identified model with parameters ρ
    calculate reconstruction error:
    if DDE case **then**
        $\varepsilon(\rho) = \|\mathbf{X}' - \Theta(\rho)\Xi(\rho)\|_2$
    else(RE case)
        $\varepsilon(\rho) = \|X - \Theta(\rho)\Xi(\rho)\|_2$
    end if
    return $\varepsilon(\rho)$
end function
external parameter optimisation
use optimisation algorithm (PS, bayesian optimisation, etc.):
$\rho^* = \arg\min_\rho$ Distributed_SINDy_Error(ρ)
compute final sparse coefficient matrix Ξ^* using ρ^*
return ρ^*, Ξ^*

Bibliography

[1] "A Neural Network Approach to Learn Delay Differential Equations via Pseudospectral Collocation." In: *ECCOMAS CONGRESS 2024, 9th European Congress on Computational Methods in Applied Sciences and Engineering*. Conference talk. Portugal, 2024.

[2] "A Neural Network Approach to Learn Delay Differential Equations via Pseudospectral Collocation." In: *PINN-PAD: Physics Informed Neural Networks in Padova*. Conference talk. Italy, 2024.

[3] A. Andò, D. Breda, G. Gava, et al. "How fast is the linear chain trick? A rigorous analysis in the context of behavioral epidemiology." In: *MBE* 17.5 (2020), pp. 5059–5084.

[4] O. Babasola, E. O. Omondi, K. Oshinubi, and N. M. Imbusi. "Stochastic delay differential equations: a comprehensive approach for understanding biosystems with application to disease modelling." In: *AppliedMath* 3.4 (2023), pp. 702–721.

[5] R. E. Baker and G. Röst. "Global dynamics of a novel delayed logistic equation arising from cell biology." In: *J. Nonlinear Sci.* 30.1 (2020), pp. 397–418.

[6] F. M. Bandi and P. C. B. Phillips. "Fully nonparametric estimation of scalar diffusion models." In: *Econometrica* 71.1 (2003), pp. 241–283.

[7] C. Bauch, A. d'Onofrio, and P. Manfredi. "Behavioral epidemiology of infectious diseases: an overview." In: *Modeling the interplay between human behavior and the spread of infectious diseases*. Ed. by P Manfredi and A d'Onofrio. Mathematical and Computational biology. New York: Springer-Verlag, 2013, pp. 1–19.

[8] A. Bellen, S. Maset, M. Zennaro, and N. Guglielmi. "Recent trends in the numerical solution of retarded functional differential equations." In: *Acta Numer.* 18 (2009), pp. 1–110.

[9] E. Beretta, D. Breda, et al. "Discrete or distributed delay? Effects on stability of population growth." In: *MBE* 13.1 (2016), pp. 19–41.

[10] J. P. Berrut and L. N. Trefethen. "Barycentric Lagrange interpolation." In: *SIAM review* 46.3 (2004), pp. 501–517.

[11] B. Boldin, O. Diekmann, and J. A. J. Metz. "Population growth in discrete time: a renewal equation•oriented survey." In: *J. Differ. Equ. Appl.* 30.8 (2024), pp. 1062–1090.

[12] L. Boninsegna, F. Nüske, and C. Clementi. "Sparse learning of stochastic dynamical equations." In: *J. Chem. Phys.* 148.24 (2018).

[13] M. R. Bonyadi and Z. Michalewicz. "Particle swarm optimization for single objective continuous space problems: a review." In: *Evol. Comput.* 25.1 (2017), pp. 1–54.

[14] F. Borgioli, D. Hajdu, T. Insperger, G. Stepan, and W. Michiels. "Pseudospectral method for assessing stability robustness for linear time-periodic delayed dynamical systems." In: *Int. J. Numer. Methods Eng.* 121.16 (2020), pp. 3505–3528.

[15] I. Bouros, R. N. Thompson, D. J. Gavaghan, and B. Lambert. "The time-dependent reproduction number for epidemics in heterogeneous populations." In: *J. R. Soc. Interface* 22.228 (2025), p. 20250095.

[16] E. Bozzo, D. Breda, and M. Tanveer. "Sparse identification of time delay systems via pseudospectral collocation." In: *IFAC-PapersOnLine* 58.27 (2024), pp. 108–113.

[17] D. Breda. *Controlling Delayed Dynamics: Advances in Theory, Methods and Applications*. Vol. 604. Springer, 2022.

[18] D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Stochastic Delay Differential Equations." In: *J. Comput. Appl. Math.* (2025), p. 117247.

[19] D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *IDEAS2024 - Investigating Dynamics in Engineering and Applied Science*. Hungary, 2024.

[20] D. Breda, D. Conte, R. D'Ambrosio, I. Santaniello, and M. Tanveer. "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *Summer school NUMRAD24 - Numerical Methods for Random Differential Models*. Switzerland, 2024.

[21] D. Breda, O. Diekmann, W. de Graaf, A. Pugliese, and R. Vermiglio. "On the formulation of epidemic models (an appraisal of Kermack and McKendrick)." In: *J. Biol. Dyn.* 6.2 (2012), pp. 103–117.

[22] D. Breda, O. Diekmann, M. Gyllenberg, F. Scarabel, and R. Vermiglio. "Pseudospectral discretization of nonlinear delay equations: new prospects for numerical bifurcation analysis." In: *SIAM J. Appl. Dyn. Syst.* 15.1 (2016), pp. 1–23.

[23] D. Breda, O. Diekmann, M. Gyllenberg, F. Scarabel, and R. Vermiglio. "Pseudospectral discretization of nonlinear delay equations: new prospects for numerical bifurcation analysis." In: *SIAM J. Appl. Dyn. Syst.* 15.1 (2016), pp. 1–23.

[24] D. Breda, O. Diekmann, D. Liessi, and F. Scarabel. "Numerical bifurcation analysis of a class of nonlinear renewal equations." In: *Electron. J. Qual. Theory Differ. Equ.* 2016.65 (2016), pp. 1–24.

[25] D. Breda, X. Ji, G. Orosz, and M. Tanveer. "Data-driven methods for delay differential equations." In: *Delays and Structures in Dynamical Systems: Modelling, Analysis and Numerical Methods*. Ed. by D. Breda, R. Vermiglio, and J. Wu. CISM Lecture Notes. Submitted. Springer, 2025. arXiv: `2512.04894`. URL: https://arxiv.org/abs/2512.04894.

[26] D. Breda and D. Liessi. "Approximation of eigenvalues of evolution operators for linear coupled renewal and retarded functional differential equations." In: *Ric. Mat.* 69.2 (2020), pp. 457–481.

[27] D. Breda, S. Maset, and R. Vermiglio. "Pseudospectral differencing methods for characteristic roots of delay differential equations." In: *SIAM Journal on Scientific Computing* 27.2 (2005), pp. 482–495.

[28] D. Breda, M. Tanveer, and J. Wu. *Sparse identification of delay equations with distributed memory*. 2025. arXiv: `2512.21070 [math.DS]`. URL: `https://arxiv.org/abs/2512.21070`.

[29] D. Breda, M. Tanveer, J. Wu, and X. Zhang. *Exploring Memory Effects: Sparse Identification in Vector-Borne Diseases*. 2026. arXiv: `2601.20591 [math.DS]`. URL: `https://arxiv.org/abs/2601.20591`.

[30] S. L. Brunton and J. N. Kutz. *Data-driven Science and Engineering*. Cambridge University Press, 2022.

[31] S. L. Brunton, J. L. Proctor, and J. N. Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems." In: *Proc. Natl. Acad. Sci.* 113.15 (2016), pp. 3932–3937.

[32] E. Buckwar. "Introduction to the numerical analysis of stochastic delay differential equations." In: *J. Comput. Appl. Math.* 125.1-2 (2000), pp. 297–307.

[33] B. Buonomo, E. Messina, and C. Panico. "Minimal epidemic models with information index: from compartmental to integral formulation." In: *Boll. Unione Mat. Ital.* (2025), pp. 1–22.

[34] B. Buonomo, E. Messina, C. Panico, and A. Vecchio. "An integral renewal equation approach to behavioural epidemic models with information index." In: *J. Math. Biol.* 90.1 (2025), p. 8.

[35] W. Cao, Z. Zhang, and G. E. Karniadakis. "Numerical Methods for Stochastic Delay Differential Equations Via the Wong–Zakai Approximation." In: *SIAM J. Sci. Comput.* 37.1 (2015), A295–A318.

[36] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. "Data-driven discovery of coordinates and governing equations." In: *Proc. Natl. Acad. Sci.* 116.45 (2019), pp. 22445–22451.

[37] R. Chartrand. "Numerical differentiation of noisy, nonsmooth data." In: *Int. Sch. Res. Not.* 2011.1 (2011), p. 164564.

[38] S. Chatterjee and A. S. Hadi. *Regression analysis by example*. John Wiley & Sons, 2015.

[39] R. TQ Chen, Y. Rubanova, J. Bettencourt, and D. K Duvenaud. "Neural ordinary differential equations." In: *NeurIPS* 31 (2018).

[40] X. Chen and I. Timofeyev. "Non-parametric estimation of stochastic differential equations from stationary time-series." In: *J. Stat. Phys.* 186 (2022), pp. 1–31.

[41] C. W. Clenshaw and A. R. Curtis. "A method for numerical integration on an automatic computer." In: *Numer. Math.* 2.1 (1960), pp. 197–205.

[42] D. Conte, I. Santaniello, D. Breda, M. Tanveer, and R. D'Ambrosio. "A Matlab code for discovering governing dynamics in Stochastic Differential Equations using SINDy algorithm and Itô-Taylor based approximations." Submitted. 2025.

[43] V. DA. "The color of environmental noise." In: *Ecology* 85 (2004), pp. 1146–1152.

[44] "Data-Driven Approaches for Structured Populations." In: Invited talk. Toronto, Canada, 2025.

[45] B. der. "A theory of the amplitude of free and forced triode vibrations." In: *Radio Review* 1 (Jan. 1920), pp. 701–710, 754.

[46] O. Diekmann, P. Getto, and M. Gyllenberg. "Stability and bifurcation analysis of Volterra functional equations in the light of suns and stars." In: *SIAM J. Math. Anal.* 39.4 (2008), pp. 1023–1069.

[47] O. Diekmann, S. A. V. Gils, S. M. V. Lunel, and H. O. Walther. *Delay equations: functional-, complex-, and nonlinear analysis*. Vol. 110. Applied Mathematical Sciences. Springer Verlag, 1995.

[48] A. d'Onofrio, P. Manfredi, and E. Salinelli. "Vaccinating behaviour, information, and the dynamics of SIR vaccine preventable diseases." In: *Theor. Popul. Biol.* 71 (2007), pp. 301–317.

[49] B. Du, Y. Wang, and X. Lian. "A stochastic predator-prey model with delays." In: *Adv. Differ. Equ.* 2015 (2015), pp. 1–16.

[50] K. J. Engel, R. Nagel, and S. Brendle. *One-parameter semigroups for linear evolution equations*. Springer, 2000.

[51] O. Enqvist, F. Jiang, and F. Kahl. "A brute-force algorithm for reconstructing a scene from two projections." In: *CVPR 2011*. IEEE. 2011, pp. 2961–2968.

[52] E. Franco, O. Diekmann, and M. Gyllenberg. "Modelling physiologically structured populations: renewal equations and partial differential equations." In: *J. Evol. Equ.* 23.3 (2023), p. 46.

[53] K. Fukushima and S. Miyake. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition." In: *Competition and Cooperation in Neural Nets*. Ed. by S.-I. Amari and Michael A. Arbib. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 267–285.

[54] C. W Gardiner et al. *Handbook of stochastic methods*. Vol. 3. springer Berlin, 1985.

[55] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[56] H. Ge et al. "Combined effects of hydrological conditions and socioeconomic factors on the seasonal dynamics of severe fever with thrombocytopenia syndrome in China, 2011–2022: a modelling study." In: *Lancet Reg. Health West. Pac.* 58 (2025).

[57] W. M. Gentleman. "Implementing Clenshaw-Curtis quadrature, I methodology and experience." In: *Communications of the ACM* 15.5 (1972), pp. 337–342.

[58] D. Ghosh, A. R. Chowdhury, and P. Saha. "Multiple delay Rössler system—Bifurcation and chaos control." In: *Chaos Soliton Fract.* 35.3 (2008), pp. 472–485.

[59] K. Gopalsamy. *Stability and Oscillations in Delay Differential Equations of Population Dynamics*. Springer, 1992.

[60] S. Guillouzic, I. L'Heureux, and A. Longtin. "Small delay approximation of stochastic delay differential equations." In: *Phys. Rev. E* 59.4 (1999), p. 3970.

[61] A. Gupta and P. F. J. Lermusiaux. "Generalized neural closure models with interpretability." In: *Sci. Rep.* 13 (2023), p. 10634.

[62] J. K. Hale and S. M. Verduyn Lunel. *Introduction to Functional Differential Equations*. Springer, 1993.

[63] J. Han and C. Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning." In: *International workshop on artificial neural networks*. Springer. 1995, pp. 195–201.

[64] P. Han, Q. Guo, H. Zhang, and L. Wang. "Approximation identification for the stochastic time-delayed dynamical system." In: *Phys. A: Stat. Mech. Appl.* 654 (2024), p. 130135.

[65] Z. He et al. "Severe fever with thrombocytopenia syndrome: a systematic review and meta-analysis of epidemiology, clinical signs, routine laboratory diagnosis, risk factors, and outcomes." In: *BMC Infect. Dis.* 20.1 (2020), p. 575.

[66] D. J. Higham. "An algorithmic introduction to numerical simulation of stochastic differential equations." In: *SIAM review* 43.3 (2001), pp. 525–546.

[67] D. J. Higham, X. Mao, and A. M. Stuart. "Strong Convergence of Euler-Type Methods for Nonlinear Stochastic Differential Equations." In: *SIAM J. Numer. Anal.* 40.3 (2002), pp. 1041–1063.

[68] A. S. Householder. "A theory of steady-state activity in nerve-fiber networks: I. Definitions and preliminary lemmas." In: *Bull. Math. Biophys.* 3.2 (1941), pp. 63–69.

[69] G. E. Hutchinson. "Circular Causal Systems in Ecology." In: *Ann. N. Y. Acad. Sci.* 50.4 (1948), pp. 221–246.

[70] J. L. W. V. Jensen. "Sur les fonctions convexes et les inégalités entre les valeurs moyennes." In: *Acta Math.* 30.1 (1906), pp. 175–193.

[71] X. A. Ji, T. G. Molnár, S. S. Avedisov, and G. Orosz. "Feed-forward Neural Networks with Trainable Delay." In: *2nd Conference on Learning for Dynamics and Control*. Vol. 120. PMLR, 2020, pp. 127–136.

[72] X. A Ji, T. G Molnár, S. S Avedisov, and G. Orosz. "Learning the dynamics of time delay systems with trainable delays." In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 930–942.

[73] X. A. Ji and G. Orosz. "Learning Time Delay Systems with Neural Ordinary Differential Equations." In: *IFAC-PapersOnLine* 55.36 (2022), pp. 79–84.

[74] X. A. Ji and G. Orosz. "Learn from one and predict all: single trajectory learning for time delay systems." In: *Nonlinear Dynamics* 112.5 (2024), pp. 3505–3518.

[75] Z. Jiang and W. Zhang. "Bifurcation analysis in single-species population model with delay." In: *Sci. China Math.* 53.6 (2010), pp. 1475–1481.

[76] T. Katayama. *Subspace Methods for System Identification*. London: Springer London, 2005. ISBN: 978-1-85233-981-4.

[77] Y. Kazmerchuk, A. Swishchuk, and J. Wu. "The pricing of options for securities markets with delayed response." In: *Math. Comput. Simul.* 75.3-4 (2007), pp. 69–79.

[78] J. Kennedy and R. Eberhart. "Particle swarm optimization." In: *International Conference on Neural Networks*. Vol. 4. IEEE. 1995, pp. 1942–1948.

[79] D. Kinga, J. B. Adam, et al. "A method for stochastic optimization." In: *International conference on learning representations (ICLR)*. Vol. 5. 6. California; 2015.

[80] D. Kleinhans. "Estimation of drift and diffusion functions from time series data: A maximum likelihood framework." In: *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 85.2 (2012), p. 026705.

[81] D. Kleinhans, R. Friedrich, A. Nawroth, and J. Peinke. "An iterative procedure for the estimation of drift and diffusion coefficients of Langevin processes." In: *Phys. Lett. A* 346.1-3 (2005), pp. 42–46.

[82] P. E. Kloeden and E Platen. *Stochastic differential equations*. Springer, 1992.

[83] B. O Koopman. "Hamiltonian systems and transformation in Hilbert space." In: *PNAS* 17.5 (1931), pp. 315–318.

[84] T. A. Köpeczi-Bócz, H. Sykora, and D. Takács. "Data-driven delay identification with SINDy." In: *International Conference on Nonlinear Dynamics and Applications*. Springer. 2023, pp. 481–491.

[85] N. Krasovskii. *Problems of the Theory of Stability of Motion*. English translation, Stanford University Press, 1963.

[86] Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory*. Springer, 1998.

[87] D. Lehotzky and T. Insperger. "A pseudospectral tau approximation for time delay systems and its comparison with other weighted-residual-type methods." In: *Int. J. Numer. Methods Eng.* 108.6 (2016), pp. 588–613.

[88] B. Lenfesty, S. Bhattacharyya, and K. Wong-Lin. "Uncovering dynamical equations of stochastic decision models using data-driven SINDy algorithm." In: *Neural Comput.* (2025), pp. 1–19.

[89] M. Levine and A. Stuart. "A framework for machine learning of model error in dynamical systems." In: *Commun. Am. Math. Soc.* 2.07 (2022), pp. 283–344.

[90] D. Liessi et al. "New developments in MatCont: delay equation importer and Lyapunov exponents." In: *arXiv preprint arXiv:2504.12785* (2025).

[91] D. Liu. "A practical guide to relu." In: *Medium* (2017).

[92] L. Ljung. "Perspectives on system identification." In: *Annu. Rev. Control.* 34.1 (2010), pp. 1–12.

[93] N. MacDonald. *Time Lags in Biological Models*. Vol. 27. Lecture notes in Biomathematics. Berlin: Springer Verlag, 1978.

[94] M. C. Mackey and L. Glass. "Oscillation and chaos in physiological control systems." In: *Science* 197.4300 (1977), pp. 287–289.

[95] X. Mao. "Robustness of exponential stability of stochastic differential delay equations." In: *IEEE Trans. Autom. Control* 41.3 (1996), pp. 442–447.

[96] D. A. Messenger and D. M. Bortz. "Weak SINDy: Galerkin-based data-driven model selection." In: *MMS* 19.3 (2021), pp. 1474–1497.

[97] E. Messina, E. Russo, and A. Vecchio. "A stable numerical method for Volterra integral equations with discontinuous kernel." In: *J. Math. Anal. Appl.* 337.2 (2008), pp. 1383–1393.

[98] J. A. Metz and O. Diekmann. *The dynamics of physiologically structured populations*. Vol. 68. Springer, 2014.

[99] G. N. Mil'shtejn. "Approximate integration of stochastic differential equations." In: *Theory Probab. Appl.* 19.3 (1975), pp. 557–562.

[100] N. Mohammadi, L. V. Santoro, and V. M. Panaretos. "Nonparametric estimation for SDE with sparsely sampled paths: An FDA perspective." In: *Stoch. Process. their Appl.* 167 (2024), p. 104239.

[101] S. A. Mohammed. "Stochastic differential systems with memory: theory, examples and applications." In: *Stochastic Analysis and Related Topics VI: Proceedings of the Sixth Oslo—Silivri Workshop Geilo 1996*. Springer. 1998, pp. 1–77.

[102] B. Øksendal and B. Øksendal. *Stochastic differential equations*. Springer, 2003.

[103] N. O'Connell. "Interacting diffusions on positive definite matrices." In: *Probab. Theory Relat. Fields* 180.3 (2021), pp. 679–726.

[104] A. Pecile, N. Demo, M. Tezzele, G. Rozza, and D. Breda. "Data-driven discovery of delay differential equations with discrete delays." In: *J. Comput. Appl. Math.* 461 (2025), p. 116439.

[105] B. Peeters and G. De Roeck. "Stochastic System Identification for Operational Modal Analysis: A Review." In: *J. Dyn. Syst. Meas. Control* 123.4 (Feb. 2001), pp. 659–667. ISSN: 0022-0434. DOI: 10.1115/1.1410370.

[106] D. Piga, F. Pura, and M. Forgione. "On the adaptation of in-context learners for system identification." In: *IFAC-PapersOnLine* 58.15 (2024), pp. 277–282.

[107] M. Ragwitz and H. Kantz. "Indispensable finite time corrections for Fokker-Planck equations from time series data." In: *Phys. Rev. Lett.* 87.25 (2001), p. 254501.

[108] M. Raissi, P. Perdikaris, and G. E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." In: *J. Comput. Phys.* 378 (2019), pp. 686–707.

[109] A. Raj, U. Simsekli, and A. Rudi. "Efficient sampling of stochastic differential equations with positive semi-definite models." In: *Adv. Neural Inf. Process. Syst.* 36 (2023), pp. 64266–64305.

[110] C. E. Rasmussen. "Gaussian processes in machine learning." In: *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[111] W. E. Ricker. "Stock and recruitment." In: *J. Fish. Res. Board Can.* 11.5 (1954), pp. 559–623.

[112] H. Risken. "Fokker-planck equation." In: *The Fokker-Planck equation: methods of solution and applications*. Springer, 1989, pp. 63–95.

[113] N. Rosli, A. Bahar, S. H. Yeak, and X. Mao. "A systematic derivation of stochastic Taylor methods for stochastic delay differential equations." In: *Bull. Malays. Math. Sci. Soc.* 36.3 (2013).

[114] N. Rosli, A. Bahar, S. H. Yeak, and H. A. Rahman. "Stochastic Taylor Methods for Stochastic Delay Differential Equations." In: *MATEMATIKA* (2013), pp. 241–251.

[115] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. "Data-driven discovery of partial differential equations." In: *Sci. adv.* 3.4 (2017), e1602614.

[116] A. Sandoz, V. Ducret, G. A Gottwald, G. Vilmart, and K. Perron. "SINDy for delay-differential equations: application to model bacterial zinc response." In: *Proc. R. Soc. Lond. A* 479.2269 (2023), p. 20220556.

[117] H. Schaeffer and S. G. McCalla. "Sparse model selection via integral terms." In: *Phys. Rev. E* 96.2 (2017), p. 023302.

[118] P. J Schmid. "Dynamic mode decomposition of numerical and experimental data." In: *J. Fluid Mech.* 656 (2010), pp. 5–28.

[119] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. "Taking the human out of the loop: A review of Bayesian optimization." In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[120] L. P. Shayer and S. A. Campbell. "Stability, bifurcation, and multistability in a system of two coupled neurons with multiple time delays." In: *SIAM J. Appl. Math.* 61.2 (2000), pp. 673–700.

[121] Y. Shi and R. Eberhart. "A modified particle swarm optimizer." In: *IEEE International Conference on Evolutionary Computation*. IEEE. 1998, pp. 69–73.

[122] H. Smith. *An Introduction to Delay Differential Equations with Applications to the Life Sciences*. Springer, 2011.

[123] "Sparse Identification of a Distributed Delay Model for a Tick-Borne Disease." In: Invited talk. Toronto, Canada, 2025.

[124] "Sparse Identification of Dynamical Systems with Distributed Delays." In: *IFAC Joint Conference on System Structure and Control (SSSC), Time Delay Systems (TDS), and Control of Complex Systems (COSY)*. Conference talk. France, 2025.

[125] "Sparse Identification of Dynamical Systems with Distributed Delays." In: *DSABNS 2025, 16th Dynamical Systems applied on Biology and Natural Sciences*. Conference talk. Italy, 2025.

[126] "Sparse Identification of Nonlinear Dynamics for Delay and Stochastic Differential Equations." In: *SciCADE 2024, International Conference on Scientific Computation and Differential Equations*. Conference talk. Singapore, 2024.

[127] "Sparse Identification of Time Delay Systems via Pseudospectral Collocation." In: *TDS 2024, 18th IFAC Workshop on Time Delay System*. Conference talk. Italy, 2024.

[128] J. C. Sprott. "A simple chaotic delay differential equation." In: *Physics Letters A* 366.4-5 (2007), pp. 397–402.

[129] L. State and P. V. Aceituno. "Training delays in spiking neural networks." In: *International Conference on Artificial Neural Networks*. Springer. 2019, pp. 713–717.

[130] S. H Streipert and G. SK Wolkowicz. "Technique to derive discrete population models with delayed growth." In: *J. Biol. Dyn.* 17.1 (2023), p. 2244987.

[131] T. Tian, K. Burrage, P. M. Burrage, and M. Carletti. "Stochastic delay differential equations for genetic regulatory networks." In: *J. Comput. Appl. Math.* 205.2 (2007), pp. 696–707.

[132] R. Tibshirani. "Regression shrinkage and selection via the lasso." In: *J. R. Stat. Soc. Series B* 58.1 (1996), pp. 267–288.

[133] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.

[134] L. N. Trefethen. "Approximation theory and approximation practice." In: *Book of abstracts*. 2009, p. 21.

[135] N. G. V. Kampen. *Stochastic processes in physics and chemistry*. Vol. 1. Elsevier, 1992.

[136] H. U Voss, J. Timmer, and J. Kurths. "Nonlinear dynamical system identification from uncertain and indirect measurements." In: *IJBC* 14.06 (2004), pp. 1905–1933.

[137] Y. Wang et al. "Data-driven discovery of stochastic differential equations." In: *Engineering* 17 (2022), pp. 244–252.

[138] G. Wanner and E. Hairer. *Solving ordinary differential equations II*. Vol. 375. Springer Berlin Heidelberg New York, 1996.

[139] M. Wanner and I. Mezić. "On higher order drift and diffusion estimates for stochastic SINDy." In: *SIAM J. Appl. Dyn. Syst.* 23.2 (2024), pp. 1504–1539.

[140] B. Wei. "Sparse dynamical system identification with simultaneous structural parameters and initial condition estimation." In: *Chaos Soliton Fract* 165 (2022), p. 112866.

[141] A. Weyer. "The Brute Force Algorithm." MA thesis. Bowling Green State University, 2019.

[142] C. K. I. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[143] J. Wu and X. Zhang. *Transmission dynamics of tick-borne diseases with co-feeding, developmental and behavioural diapause*. Springer Nature, 2020.

[144] Y. Wu. "Reconstruction of delay differential equations via learning parameterized dictionary." In: *Physica D* 446 (2023), p. 133647.

[145] Z. Wu, H. Huang, and L. Wang. "Stochastic delay logistic model under regime switching." In: *Abstract and Applied Analysis*. Vol. 2012. 1. Wiley Online Library. 2012, p. 241702.

[146] M. Xu, Y. Wei, and J. Wei. "Bifurcation analysis of Rössler system with multiple delayed feedback." In: *Electron. J. Qual. Theory Differ. Equ.* 2010.63 (2010), pp. 1–22.

[147] N. Zhang et al. "Modelling the transmission dynamics of severe fever with thrombocytopenia syndrome in Jiangsu Province, China." In: *Parasit. vectors* 14.1 (2021), p. 237.

[148] X. Zhang et al. "Patterns of Transmission Dynamics of Severe Fever with Thrombocytopenia Syndrome Virus in Dalian: Roles of Systemic, Co-Feeding, and Transovarial Routes." In: *CSIAM-TLS* 1.2 (2025), pp. 299–319. ISSN: 3006-2721.

يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ

"*Allah will raise in rank those of you who have faith and those who have been given knowledge.* "

"Al Quran", (Sūrat al-Mujādilah, No 58, Āyat 11).