

Complexity

Complex Algorithms for Data-Driven Model Learning in Science and Engineering

Lead Guest Editor: Francisco J. Montáns

Guest Editors: Francisco Chinesta, Rafael Gómez-Bombarelli, and Jose N. Kutz





Complex Algorithms for Data-Driven Model Learning in Science and Engineering

Complexity

Complex Algorithms for Data-Driven Model Learning in Science and Engineering

Lead Guest Editor: Francisco J. Montáns

Guest Editors: Francisco Chinesta, Rafael Gómez-Bombarelli,
and Jose N. Kutz



Copyright © 2019 Hindawi. All rights reserved.

This is a special issue published in “Complexity.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board


- José A. Acosta, Spain
Carlos F. Aguilar-Ibáñez, Mexico
Mojtaba Ahmadiéh Khanesar, UK
Tarek Ahmed-Ali, France
Alex Alexandridis, Greece
Basil M. Al-Hadithi, Spain
Juan A. Almendral, Spain
Diego R. Amancio, Brazil
David Arroyo, Spain
Mohamed Boutayeb, France
Átila Bueno, Brazil
Arturo Buscarino, Italy
Guido Caldarelli, Italy
Eric Campos-Canton, Mexico
Mohammed Chadli, France
Émile J. L. Chappin, Netherlands
Diyi Chen, China
Yu-Wang Chen, UK
Giulio Cimini, Italy
Danilo Comminiello, Italy
Sara Dadras, USA
Sergey Dashkovskiy, Germany
Manlio De Domenico, Italy
Pietro De Lellis, Italy
Albert Diaz-Guilera, Spain
Thach Ngoc Dinh, France
Jordi Duch, Spain
Marcio Eisencraft, Brazil
Joshua Epstein, USA
Mondher Farza, France
Thierry Floquet, France
Mattia Frasca, Italy
José Manuel Galán, Spain
Lucia Valentina Gambuzza, Italy
Bernhard C. Geiger, Austria
Carlos Gershenson, Mexico
- Peter Giesl, UK
Sergio Gómez, Spain
Lingzhong Guo, UK
Xianggui Guo, China
Sigurdur F. Hafstein, Iceland
Chittaranjan Hens, India
Giacomo Innocenti, Italy
Sarangapani Jagannathan, USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, UK
M. Hassan Khooban, Denmark
Abbas Khosravi, Australia
Toshikazu Kuniya, Japan
Vincent Labatut, France
Lucas Lacasa, UK
Guang Li, UK
Qingdu Li, China
Chongyang Liu, China
Xiaoping Liu, Canada
Xinzhi Liu, Canada
Rosa M. Lopez Gutierrez, Mexico
Vittorio Loreto, Italy
Noureddine Manamanni, France
Didier Maquin, France
Eulalia Martínez, Spain
Marcelo Messias, Brazil
Ana Meštrović, Croatia
Ludovico Minati, Japan
Ch. P. Monterola, Philippines
Marcin Mrugalski, Poland
Roberto Natella, Italy
Sing Kiong Nguang, New Zealand
Nam-Phong Nguyen, USA
B. M. Ombuki-Berman, Canada
Irene Otero-Muras, Spain
Yongping Pan, Singapore
- Daniela Paolotti, Italy
Cornelio Posadas-Castillo, Mexico
Mahardhika Pratama, Singapore
Luis M. Rocha, USA
Miguel Romance, Spain
Avimanyu Sahoo, USA
Matilde Santos, Spain
Josep Sardanyés Cayuela, Spain
Ramaswamy Savitha, Singapore
Hiroki Sayama, USA
Michele Scarpiniti, Italy
Enzo Pasquale Scilingo, Italy
Dan Selisțeanu, Romania
Dehua Shen, China
Dimitrios Stamovlasis, Greece
Samuel Stanton, USA
Roberto Tonelli, Italy
Shahadat Uddin, Australia
Gaetano Valenza, Italy
Alejandro F. Villaverde, Spain
Dimitri Volchenkov, USA
Christos Volos, Greece
Qingling Wang, China
Wenqin Wang, China
Zidong Wang, UK
Yan-Ling Wei, Singapore
Honglei Xu, Australia
Yong Xu, China
Xinggang Yan, UK
Baris Yuçe, UK
Massimiliano Zanin, Spain
Hassan Zargazadeh, USA
Rongqing Zhang, USA
Xianming Zhang, Australia
Xiaopeng Zhao, USA
Quanmin Zhu, UK

Contents

Complex Algorithms for Data-Driven Model Learning in Science and Engineering

Francisco J. Montáns , Francisco Chinesta , Rafael Gómez-Bombarelli, and J. Nathan Kutz 
Editorial (3 pages), Article ID 5040637, Volume 2019 (2019)


Analyzing Nonlinear Dynamics via Data-Driven Dynamic Mode Decomposition-Like Methods

Soledad Le Clainche and José M. Vega 
Review Article (21 pages), Article ID 6920783, Volume 2018 (2019)



Computer Vision with Error Estimation for Reduced Order Modeling of Macroscopic Mechanical Tests

Franck Nguyen, Selim M. Barhli, Daniel Pino Muñoz, and David Ryckelynck 
Research Article (10 pages), Article ID 3791543, Volume 2018 (2019)

Applied Koopman Theory for Partial Differential Equations and Data-Driven Modeling of Spatio-Temporal Systems

J. Nathan Kutz , J. L. Proctor, and S. L. Brunton
Research Article (16 pages), Article ID 6010634, Volume 2018 (2019)

Long-Time Predictive Modeling of Nonlinear Dynamical Systems Using Neural Networks

Shaowu Pan  and Karthik Duraisamy 
Research Article (26 pages), Article ID 4801012, Volume 2018 (2019)





A Damage Classification Approach for Structural Health Monitoring Using Machine Learning

Diego Tibaduiza , Miguel Ángel Torres-Arredondo, Jaime Vitola, Maribel Anaya, and Francesc Pozo 
Research Article (14 pages), Article ID 5081283, Volume 2018 (2019)






Artificial Neural Networks Based Friction Law for Elastomeric Materials Applied in Finite Element Sliding Contact Simulations

Aleksandra Serafińska, Wolfgang Graf, and Michael Kaliske 
Research Article (15 pages), Article ID 4396758, Volume 2018 (2019)

A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition

Rubén Ibáñez, Emmanuelle Abisset-Chavanne, Amine Ammar , David González , Elías Cueto , Antonio Huerta, Jean Louis Duval, and Francisco Chinesta 
Research Article (11 pages), Article ID 5608286, Volume 2018 (2019)




A Conceptual Approach to Complex Model Management with Generalized Modelling Patterns and Evolutionary Identification

Sergey V. Kovalchuk , Oleg G. Metsker, Anastasia A. Funkner , Ilia O. Kisliakovskii , Nikolay O. Nikitin, Anna V. Kalyuzhnaya , Danila A. Vaganov, and Klavdiya O. Bochenina 
Research Article (15 pages), Article ID 5870987, Volume 2018 (2019)

Data-Driven Finite Element Models of Passive Filamentary Networks

Brian Adam and Sorin Mitran 
Research Article (7 pages), Article ID 6878265, Volume 2018 (2019)

A Data-Driven Parameter Adaptive Clustering Algorithm Based on Density Peak

Tao Du , Shouning Qu , and Qin Wang 
Research Article (14 pages), Article ID 5232543, Volume 2018 (2019)

Editorial

Complex Algorithms for Data-Driven Model Learning in Science and Engineering

Francisco J. Montáns ¹, **Francisco Chinesta** ²,
Rafael Gómez-Bombarelli³ and **J. Nathan Kutz** ⁴

¹*Escuela Técnica Superior de Ingenieros Aeronáuticos, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, E-28045 Madrid, Spain*

²*ESI Chair, École Nationale Supérieure d'Arts et Métiers ParisTech, 151 bd. De L'Hôpital, F-75013 Paris, France*

³*Department of Materials Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA*

⁴*Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA*

Correspondence should be addressed to Francisco J. Montáns; fco.montans@upm.es

Received 22 May 2019; Accepted 22 May 2019; Published 9 June 2019

Copyright © 2019 Francisco J. Montáns et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In their first centuries, scientific and engineering developments were dominated by empirical understanding which encapsulated the first paradigm of scientific discovery. After the Renaissance, the scientific revolution and the development of calculus led to a new scientific viewpoint whereby physical principles, laws of nature, and engineering models were established by proposing new theoretical constructs that could be verified through specific experiments. This was the second paradigm of scientific discovery. More recently, the computational era, or the third paradigm of discovery, has allowed us to solve complex and nonlinear scientific and engineering problems that were beyond our analytically tractable methodologies. Today, there is a new fourth paradigm of discovery, which is a data-driven science and engineering framework whereby complex models and physical laws are directly inferred from data.

Therefore, there is increasing change in the objective of computational algorithms used in simulations. Until now, the purpose was to accurately discretize systems of linear and nonlinear continuum equations derived from physical laws, models, and principles frequently established prior to the computational era; these equations were inferred from observation on limited experimental data and significantly simplified to make them analytically tractable. Today, the available experimental data and the complexity of the equations are no longer a major limitation to the point that we may

compute physical processes without resorting to analytical laws, principles, or models; we just need to predict the correct output from the system for a given input even when there is not a well-defined model. However, for this endeavor, we need new computational algorithms capable of learning the complex behavior of the system and of establishing those governing equations of the system directly from experimental data, with the flexibility of not having to rely on analytical equations. An example is the determination of the nonlinear behavior of solids and fluids under general conditions directly from measured data, without specifying the form of the constitutive relations. Many fields already have started to capitalize on such methods, developing algorithms for fuzzy relations, leading to data-driven decision-making by constructing purely computational predictive analytics in such complex fields as economics, consumer behavior and dynamics, security, and even web utilities. The engineering sciences are now poised to also take advantage of data-driven methods in obtaining physical principles and models which yield reliable laws and accurate predictions, using fewer hypotheses and fewer analytical relations and balancing the parametrization of physical models with the amount of available measurements.

The purpose of the special issue on data-driven model learning in science and engineering is to bring representative novel state-of-the-art contributions in this line. Many papers

were submitted from leading authors in data-driven procedures. After a rigorous review process, among them, only ten outstanding contributions were selected for the special issue. These are representative of different algorithmic approaches and application fields.

Clustering is an important aspect of unsupervised data-driven procedures. In these techniques, data groups are identified and tagged. T. Du et al., in their paper entitled “A Data-Driven Parameter Adaptive Clustering Algorithm Based on Density Peak,” develop such an algorithm named DDPA-DP to avoid the influence of artificial parameters, improving the congruence and flexibility of clustering. The authors compare their proposal to other existing algorithms, using both synthetic data and real-world data obtained from the thermal power industry. The authors conclude that their proposal is advantageous in terms of accuracy of clustering and time complexity.

Reduced order modeling will be extremely important in biomechanics, for example, in obtaining workable constitutive relations of complex phenomena for simulations of tissue and cell behavior. In this line, B. Adam and S. Mitran present in “Data-Driven Finite Element Models of Passive Filamentary Networks” a procedure to substantially reduce the cost of finite element simulations of actin filaments in the cytoskeleton. They employ at the smaller scale Brownian dynamics with tens of millions of state variables. Then, applying a singular value decomposition, they reduce the problem to obtain a representative numerical constitutive equation at a much smaller scale. They show through an example that the reduced model captures the key features present in the behavior of the full model.

Data-driven algorithms have been used for long in the field of computer vision and are now being increasingly used in mechanics and material science. F. Nguyen et al., in their paper “Computer Vision with Error Estimation for Reduced Order Modeling of Macroscopic Mechanical Tests,” use clustering and convolutional neural networks trained by supervised machine learning on digital images of mechanical tests on a specimen, under a variety of loading conditions. With their procedure, they identify a reduced order model. This reduced order model is fast to be employed in manufacturing procedures to make part-specific decisions.

Data-driven models bring a new ingredient in the overall modeling of complex systems. Data-driven models may be combined with classical modeling approaches and ensemble-based modeling. Therefore, new meta-modeling methods are also needed to manage, in a systematic and automatic manner, the combined modeling and application procedures themselves. S. V. Kovalchuk et al., in their paper “A Conceptual Approach to Complex Model Management with Generalized Modelling Patterns and Evolutionary Identification,” present such an algorithm for mixed model management, considering the spaces of functions, parameters, and modeling approaches relating to them. They show in their paper interesting applications in metocean simulations, health care processes, and mining in social media.

One of the main aspects in the design of tires is the friction behavior, which largely affects its performance, temperature, and longevity. The friction laws are complex

functions of different parameters like sliding velocity, contact pressure, and temperature. The usual approach is to establish an analytical law with fitted material parameters (e.g., the Huemer friction law). A more flexible and accurate data-driven approach is presented by A. Serafińska et al. in their paper entitled “Artificial Neural Networks Based Friction Law for Elastomeric Materials Applied in Finite Element Sliding Contact Simulations.” In their work, the authors use artificial neural networks to obtain a regularized numerical nonlinear data-driven thermomechanical friction law, function of the mentioned variables, and the temperature. They obtain an excellent fitting to experimental results. They include their law in a finite element contact formulation and perform simulations in tires for different acceleration/brake conditions.

In the aircraft industry, the timely detection of internal damage in composite structures is a difficult and complex procedure. The purpose of structural health monitoring (SHM) is to be able to perform such detections and damage classification. In this field, data-driven approaches are especially useful and very promising. The paper of Tibaduiza et al. entitled “A Damage Classification Approach for Structural Health Monitoring Using Machine Learning” presents a data-driven methodology using data collected from piezoelectric sensors under different structural states and guided waves to identify damage type and location in CFRP (carbon fibre-reinforced polymer) sandwich structures and plates. Their examples include different types of damage as delamination and cracking of the skin. Their procedure consists in a hierarchical nonlinear principal component analyses with machine learning.

S. Pan and K. Duraisamy, in their paper “Long-Time Predictive Modeling of Nonlinear Dynamical Systems Using Neural Networks,” develop data-driven models for nonlinear dynamical systems using feedforward neural networks with a Jacobian regularization for the loss function. The purpose of the Jacobian regularization is, for example, to improve the robustness of the model for limited data and to improve the predictions when the model is unstable. They compared their approach to a sparse identification of nonlinear dynamical systems approach with a background function library.

Also in the field of nonlinear dynamic analysis of systems, in the paper “Analyzing Nonlinear Dynamics via Data-Driven Dynamic Mode Decomposition-Like Methods,” S. Le Clainche and J. M. Vega review and analyze two different approaches useful for data-driven analyses. These approaches are the high-order dynamic mode decomposition (HODMD) method, based on the classical dynamic mode decomposition, and the spatiotemporal Koopman decomposition (STKD) by linear expansion. In a nutshell, the former considers different snapshots (steps) to account for the evolutionary, nonlinear response of the system through different updating matrices, whereas the second approach considers different spatiotemporal decomposition (possibly nonlinear time frequency and spatial wave number correlations of the components). Some applications are also explored in their paper.

Koopman representations, with more general Koopman eigenfunctions, are also used by J. N. Kutz et al. in their paper “Applied Koopman Theory for Partial Differential Equations

and Data-Driven Modeling of Spatio-Temporal Systems.” The authors focus their presentation in the impact of the choice of observable variables in the quality of the approximations obtained. They perform the analysis through several carefully selected examples which highlight their conclusions; these examples are the Burgers equation, the nonlinear Schrödinger equation, the cubic-quintic Ginzburg-Landau equation, and the equations from a reaction-diffusion system. They demonstrate that a poor choice of the observable variables brings approximations worse than those obtained with the classical DMD. The authors present a step-by-step procedure for a good selection of the mentioned variables.

R. Ibáñez and his coworkers present in their paper “A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition” a novel PGD-based data-driven identification procedure for high dimensional problems using Kriging interpolants. This procedure uses nonstructured datasets, is robust with respect to high dimensionality, and alleviates the curse of dimensionality, this latter property obtained through the method of separation of variables employed by the authors in previous publications to obtain reduced order models. To avoid nonparsimonious predictions, due to the solution being in a highly nonlinear manifold, the authors use sliced domains, collocation points, and local PGD versions. They name the method “sparse-PGD.” They compare their approach to other available solution methods using different synthetic and physical problems.

Conflicts of Interest

The editors of the special issue declare that they have no conflicts of interest regarding the publication of this special issue.

Acknowledgments

The guest editorial team would like to thank the authors of the submitted papers for their interest on the special issue and for their valuable contributions; and the anonymous reviewers for their careful work, their advice to the editors, and their suggestions to improve the manuscripts. We also thank the publishing team for their assistance and support in all the process.

Francisco J. Montáns
Francisco Chinesta
Rafael Gómez-Bombarelli
J. Nathan Kutz

Review Article

Analyzing Nonlinear Dynamics via Data-Driven Dynamic Mode Decomposition-Like Methods

Soledad Le Clainche and José M. Vega 

E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to José M. Vega; josemanuel.vega@upm.es

Received 12 June 2018; Revised 11 September 2018; Accepted 24 October 2018; Published 12 December 2018

Academic Editor: J. Nathan Kutz

Copyright © 2018 Soledad Le Clainche and José M. Vega. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article presents a review on two methods based on dynamic mode decomposition and its multiple applications, focusing on higher order dynamic mode decomposition (which provides a purely temporal Fourier-like decomposition) and spatiotemporal Koopman decomposition (which gives a spatiotemporal Fourier-like decomposition). These methods are purely data-driven, using either numerical or experimental data, and permit reconstructing the given data and identifying the temporal growth rates and frequencies involved in the dynamics and the spatial growth rates and wavenumbers in the case of the spatiotemporal Koopman decomposition. Thus, they may be used to either identify and extrapolate the dynamics from transient behavior to permanent dynamics or construct efficient, purely data-driven reduced order models.

1. Introduction

Uncovering the quantitative essence of complex signals, either numerical or experimental, coming from nonlinear systems is an interesting topic in *data science* and *fluid dynamics*. Among its several applications in multidisciplinary sciences, the purely data-driven analysis of nonlinear dynamics has raised the interest of applied mathematicians (to understand the dynamics of ordinary or partial differential equations and systems, such as the complex Ginzburg-Landau equation), doctors (to find relevant patterns in medical images), and aeronautical, mechanical, and civil engineers (looking for flow patterns or global instabilities or computing aerodynamic forces on slender or bluff bodies). Even in simple, low-dimensional dynamical systems, such as the Lorenz system (3 ordinary differential equations), it is possible to find complex dynamics, including periodic, quasi-periodic, and chaotic behaviors, as well as complex transients.

Fourier analysis has long been used as a data-processing tool for analyzing complex signals. In addition to improving the standard Fourier transform (e.g., *fast Fourier transform*, *power spectral density* [1], and the so-called *Laskar method* [2, 3]), several techniques have been developed in recent years that extend the scope of Fourier analysis by seeking not

only the frequencies of the signal but also the growth rates, which are relevant to transient behavior. One of the most popular methods to obtain such an expansion is *dynamic mode decomposition* (DMD) [4] (see also [5–7]), which needs to be extended to give optimal results [8], as will be discussed in this paper. These DMD-like methods are based on the idea of decomposing a signal as an expansion in Fourier-like modes, as

$$\mathbf{v}(t) \approx \sum_{n=1}^N a_n \mathbf{u}_n e^{(\delta_n + i\omega_n)t} \quad \text{for } t_1 \leq t \leq t_1 + T, \quad (1)$$

where \mathbf{v} is a time-dependent vector signal, which can be either real or complex, \mathbf{u}_n are conveniently normalized spatial modes, $a_n > 0$ are the mode amplitudes, and δ_n and ω_n are the associated growth rates and frequencies, respectively. Dynamics of the type (1) are paramount in applications, as it will be illustrated in the paper. A good example is the joint wake (accounting for interaction between the individual wakes of the various buildings) of urban topographies; see Figure 1. This example is interesting for two reasons. First, direct numerical simulation is not affordable in realistic conditions due to the huge value of the Reynolds number,

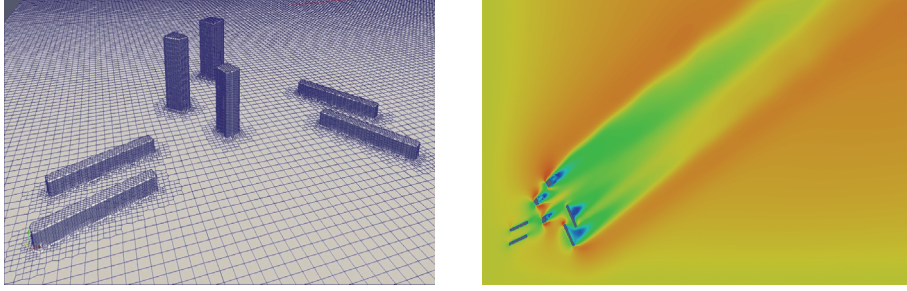


FIGURE 1: An urban topography composed by three skyscrapers and four elongated buildings (left) and horizontal snapshots of the associated joint wake (right).

meaning that numerical data must be obtained using turbulence modeling, which only captures the large scales. These are temporally quasi-periodic and may be described in the form (1); small scales, instead, exhibit spatiotemporal chaos. Experimental data, on the other hand, exhibit nonnegligible errors that must be filtered out and also describe the large scales only.

Expansion (1) is appropriate to represent temporally growing (if $\delta_n > 0$) or decaying (if $\delta_n < 0$), periodic (if the frequencies ω_n are commensurable) or quasi-periodic (if the frequencies are incommensurable) phenomena. Commensurability is a subtle matter in finite precision computations, where exact incommensurability is not possible. Instead, we shall guess that a set of frequencies are incommensurable when the computed values are not commensurable to some small tolerance. Namely, e.g., three frequencies are incommensurable if no relation of the form $m\omega_1 \pm n\omega_2 \pm p\omega_3 = \varepsilon$ is possible for moderate values (say, not larger than 10) of the natural numbers m , n , and p and some small (say, $\sim 10^{-5}$) ε . This can be discerned provided that the frequencies are computed with great accuracy. Moreover, the robustness and accuracy of the approximation (1) are a good means to ensure that the right periodic or quasi-periodic dynamics have been captured. In chaotic dynamics, instead, an infinite number of modes should be considered even for moderate accuracy and the analysis using these methods is subtle.

In the data-driven methods described and used along the paper, space and time will both be discretized. However, it is enlightening to consider at the moment continuous time, with the vector snapshots \mathbf{v} of size J . The simplest problem giving the dynamics in the right hand side of (1) is the ODE linear system

$$\frac{d\mathbf{v}(t)}{dt} = \mathbf{A}\mathbf{v}(t), \quad (2)$$

for an appropriate $J \times J$ matrix \mathbf{A} that may be computed via the pseudoinverse (applied to a discretized version of this equation, considering a collection of nearby snapshots and approximating the time derivative by finite differences).

Obviously, (2) makes sense if the dynamics associated with the snapshots are linear. However, expansion (1) and thus the linear equation (2) also apply to nonlinear dynamics. This is because the linear system (2) only intends to approximate

the particular trajectory in the left hand side of (1), not the complete dynamics of the nonlinear system. In other words, if a different trajectory of the nonlinear system is given, for different initial conditions, matrix \mathbf{A} is also generally different. In any event, for generic \mathbf{A} , all solutions of (2) are of the form displayed in the right hand side of (1), where the modes \mathbf{u}_n are the eigenvectors of \mathbf{A} and $\delta_n + i\omega_n$ are the associated eigenvalues. However, not all expansions of form (1) are solutions of (2) because (2) provides at most J modes, while the number of terms involved in (1), N , which is known as the *spectral complexity* of expansion (1), can be larger than J . More precisely, defining the *spatial complexity* of expansion (1), $M \leq J$, as the dimension of the span of the modes $\mathbf{u}_1, \dots, \mathbf{u}_N$, the linear problem (2) is appropriate to compute (1) only if $M = N$ (note that M cannot be larger than N). We insist here that the spectral and spatial complexities are defined for expansion (1), not for the dynamical system from which the snapshots may come. However, $N > M$ in many dynamical systems of scientific/technological interest, such as the following:

- (i) The *Lorenz system* [9] (and other ODEs) in which $M = 3$, while N may be quite large
- (ii) Many *nonlinear, infinite-dimensional dynamical systems* (PDEs) exhibiting an *inertial manifold* [10] (limited M) but complex (e.g., quasi-periodic) temporal behavior
- (iii) *Extrapolating* from transients to obtain attractors [11] requiring identifying additional (temporally decaying modes), which increases N
- (iv) *Experimental noise/numerical errors* adding (unphysical) temporal complexity that increases N and must be identified to *clean the postprocessed data*

When $N > M$, the number of involved modes is increased by using a higher order ODE system. Specifically the linear system (1) can be replaced by

$$\frac{d^d \mathbf{v}}{dt^d} = \mathbf{A}_1 \frac{d^{d-1} \mathbf{v}(t)}{dt^{d-1}} + \mathbf{A}_2 \frac{d^{d-2} \mathbf{v}(t)}{dt^{d-2}} + \dots + \mathbf{A}_{d-1} \frac{d\mathbf{v}(t)}{dt} + \mathbf{A}_d \mathbf{v}(t), \quad (3)$$

for appropriate $J \times J$ matrices $\mathbf{A}_1, \dots, \mathbf{A}_d$. Obviously, this linear problem can also be written in the form (3), as

$$\frac{d\bar{\mathbf{v}}}{dt} = \mathbf{A}\bar{\mathbf{v}}. \quad (4)$$

Here, $\bar{\mathbf{v}}$ is a vector of size Jd that collects \mathbf{v} , $d\mathbf{v}/dt, \dots$, and $d^{d-1}\mathbf{v}/dt^{d-1}$, and the $(Jd) \times (Jd)$ -matrix \mathbf{A} is a sparse matrix whose blocks are the unit $J \times J$ -unit matrix and the matrices $\mathbf{A}_1, \dots, \mathbf{A}_d$. The matrix \mathbf{A} can be computed via the pseudoinverse applied to $\bar{\mathbf{v}}(t)$. Now, for an appropriate value of the index d , any dynamics of the form displayed in the right hand side of (1) can be captured. Note that the size of the matrix \mathbf{A} is huge when J and d are both large, which would require very large computational resources. However, these resources are greatly decreased by applying this idea after appropriate dimension reduction; see below.

In the discrete counterpart of (1), the time variable is discretized at K equispaced values of t , as $t_k = t_1 + (k-1)\Delta t$, with $\Delta t = T/K$, and a set of snapshots is defined as $\mathbf{v}_k = \mathbf{v}(t_k)$. The counterparts of (1), (2), and (3) are

$$\mathbf{v}_k \simeq \sum_{n=1}^N a_n \mathbf{u}_n e^{(\delta_n + i\omega_n)t_k} \quad \text{for } k = 1, \dots, K, \quad (5)$$

$$\mathbf{v}_{k+1} = \mathbf{R}^t \mathbf{v}_k \quad \text{for } k = 1, \dots, K-1, \quad (6)$$

$$\mathbf{v}_{k+d} = \mathbf{R}_1^t \mathbf{v}_{k+d-1} + \mathbf{R}_2^t \mathbf{v}_{k+d-2} + \dots + \mathbf{R}_d^t \mathbf{v}_k \quad (7)$$

for $k = 1, \dots, K-d$,

respectively. Obviously, (1) follows from (5) by just setting t_k continuous (interpolation in t). Interpolation can also be made in the spatial variable, which yields the continuous *temporal DMD* expansion (cf. (1)):

$$\mathbf{v}(\mathbf{x}, t) \simeq \sum_{n=1}^N a_n \mathbf{u}_n(\mathbf{x}) e^{(\delta_n + i\omega_n)t} \quad \text{for } t_1 \leq t \leq t_1 + T, \quad (8)$$

where the modes \mathbf{u}_n depend on the spatial variable and the vector \mathbf{v} may account for a vector state variable, such as the velocity vector in fluid flows.

Equation (6) is the essence of *standard DMD* [6] that, as explained above, only applies when the spatial and spectral complexities coincide. Equation (7), instead, is the basis of *higher order DMD* (HODMD) [8]. This method permits computing the expansion (1) in the general case $M < N$ choosing an appropriate value of the tunable index d . Note that the main difference between (6) and (7) is that the latter equation relates each snapshot with not only the last snapshot but also the former $d-1$ snapshots. In fact, HODMD reduces to standard DMD if $d = 1$. For $d > 1$, instead, HODMD can be seen as a result of applying standard DMD to a set of enlarged snapshots containing also the delayed snapshots. Thus, HODMD synergically combines the advantages of standard DMD and some consequences [12] of the delayed embedding theorem by Takens [13], who followed and formalized former seminal ideas by Packard et al. [14]. In fact, combination of DMD and delayed snapshots had previously been suggested [7] and performed [15], in a

spirit different from that in [8]. Other applications of 'Takens' theorem in different contexts (not related to DMD) include, e.g., modal parameter identification and model reduction [16], identification of invariant sets and bifurcations from time-series [17], and analysis of high-dimensional time-series combining the Laplacian eigenmaps with time-lagged embedding [18]. On the other hand, some of the many improvements of standard DMD based on different ideas (not related to time-lagged snapshots) are as follows:

- (i) Optimized DMD [19], in which the expansion is computed via an optimization problem. Compared to this method, standard DMD and HODMD are simpler, purely linear algebra driven approaches. However, some linear algebra oriented (and thus less computationally expensive) approaches have been developed to treat separable nonlinear least squares problems [20], in particular, that involved in optimized DMD [21].
- (ii) Sparsity promoting DMD [22], which uses a penalty to identify a smaller set of important modes via convex optimization techniques.
- (iii) Extended DMD [23], which extends the standard DMD approximation to include more basis functions, allowing the method to capture more complex behavior.
- (iv) Total least squares DMD [24], which combines total least squares and standard DMD to treat noisy data. Let us mention here that other improvements of standard DMD have been developed to correct the effect of noise. Among these, Dawson et al. [25] also tried to characterize the noise properties, Takeishi et al. [26] combined a Bayesian formulation with DMD, and Dicle et al. [27] computed low-rank factors of the DMD operator while satisfying the total least squares constraints.

On the other hand, for spatially developing flows, t may be replaced in (8) by a distinguished spatial variable x , obtaining the *spatial DMD expansion*

$$\mathbf{v}(x, \mathbf{y}, t) = \sum_{m=1}^M a_m \mathbf{u}_m(\mathbf{y}, t) e^{(\nu_m + i\kappa_m)x}, \quad (9)$$

where \mathbf{y} collects the remaining transversal spatial variables and ν_m and κ_m are the spatial growth rates and wavenumbers, respectively. Still, a simultaneous exponential behavior in space and time may be relevant, considering the *spatiotemporal Koopman decomposition* (STKD) [28]

$$\mathbf{v}(x, \mathbf{y}, t) = \sum_{m=1}^M \sum_{n=1}^N a_{mn} \mathbf{u}_{mn}(\mathbf{y}) e^{(\nu_m + i\kappa_m)x + (\delta_n + i\omega_n)t}, \quad (10)$$

which will be obtained by an appropriate STKD method described below. This method roughly consists in applying HODMD along the x and t coordinates simultaneously. Expansion (10) has been attempted in [29] in a data-driven fashion, through an approach based on Koopman operator

theory plus the use of appropriate basis functions. See also the analysis in [30], where some aspects of the theory of operator-valued kernels for machine learning with delay-coordinate maps of dynamical systems are combined. Similar spatiotemporal expansions have also been already derived (via a not purely data-driven method) for the Navier-Stokes (NS) equations in [31] by first projecting the NS equations into the temporal modes and then applying spatial DMD to these modes using appropriate basis functions.

The modes $\mathbf{u}_{mn}(\mathbf{y})$ are appropriately normalized and expansion (10) will be further truncated by retaining only those modes whose amplitudes a_{mn} are larger than a given threshold. In other words, expansion (10) will not generally contain the $M \times N$ modes indicated in (10), but a smaller number of modes, which is emphasized rewriting (10) as

$$\mathbf{v}(x, \mathbf{y}, t) = \sum_{m,n} a_{mn} \mathbf{u}_{mn}(\mathbf{y}) e^{(v_m + i\kappa_m)x + (\delta_n + i\omega_n)t}. \quad (11)$$

It is interesting to note that the methods considered above are related to former seminal ideas by Koopman [32]; see also [5, 31, 33]. *Koopman operator theory* is concerned with *observability* in dynamical systems. Specifically, for a dynamical system $\mathbf{v}(t)$, we may consider the temporal shift $t \rightarrow t + \Delta t$, with fixed Δt . For an arbitrary vector observable function \mathbf{g} , the Koopman operator \mathbf{R}^\dagger is defined as

$$(\mathbf{R}^\dagger \cdot \mathbf{g})(\mathbf{v}(t)) = \mathbf{g}(\mathbf{v}(t + \Delta t)). \quad (12)$$

This operator is linear, because obviously $\mathbf{R}^\dagger \cdot (\alpha_1 \mathbf{g}_1 + \alpha_2 \mathbf{g}_2) = \alpha_1 \mathbf{R}^\dagger \cdot \mathbf{g}_1 + \alpha_2 \mathbf{R}^\dagger \cdot \mathbf{g}_2$, and infinite-dimensional, since it acts on the infinite-dimensional space of observables. Thus, its eigenvalues and eigenvectors allow for giving an exact representation of any observable. Setting \mathbf{g} = the identity, the Koopman operator (12) reduces to (6) and is thus related to the standard DMD method. Similarly, considering an observable that also takes into account delayed snapshots, the Koopman operator reduces to (7) and is thus related to HODMD. Still, particular realizations of the spatiotemporal Koopman operator $\mathbf{R}^{x,t}$, defined as

$$(\mathbf{R}^{x,t} \cdot \mathbf{g})(\mathbf{v}(x, \mathbf{y}, t)) = \mathbf{v}(x + \Delta x, \mathbf{y}, t + \Delta t), \quad (13)$$

are related to the STKD method mentioned above. This connection is why we call this method spatiotemporal Koopman decomposition.

Finally, we note that in the weakly nonlinear analysis of oscillatory pattern forming systems [34, 35], some equations similar to (9) and (10) are also obtained. However, these methods require both knowing the governing equations and assuming that the dynamics are small perturbations of a known steady or periodic solution. In contrast, DMD, HODMD, and STKD provide approximations of type (9) and (10) in a purely data-driven fashion for arbitrary fully nonlinear dynamics, which are great advantages, specially in the case of the analysis of experimental data. Note in this context that expansions (9) and (10) can be seen as *semianalytical expressions* that allow for the fast online computation of the associated spatiotemporal data, which leads to a *purely data-driven reduced order model* (ROM). Among the most

important applications of such a ROM, (i) it avoids numerical computation, which may be quite computationally expensive in large scale systems, or even not possible when the model is obtained from experimental data; (ii) it predicts different flow states through available data reducing the number of data collected in experiments; and (iii) it provides efficient tools for various applications, including flow control [36] and optimization design [37].

With the above in mind, the main object of this paper is to present the HODMD and STKD methods, illustrating them in simple toy models and applying them to various problems of scientific and technological interest. To this end, the HODMD method is described and illustrated in a simple toy model in Section 2, while several applications of this method are considered in Section 3. The STKD method is developed in Section 4 and applied to various problems in Section 5. The paper ends with some concluding remarks in Section 6.

2. Higher Order Dynamic Mode Decomposition

As above, for a time-dependent dynamical system given by the vector state variable $\mathbf{v}(t)$, of size J , we consider a set of K snapshots at equispaced values of t ,

$$\mathbf{v}_k \equiv \mathbf{v}(t_k), \quad (14)$$

$$\text{with } t_k = t_1 + (k - 1) \Delta t \text{ for } k = 1, \dots, K.$$

If the dynamical system is infinite-dimensional, the finite-dimensional state vector \mathbf{v} generally results from spatial discretization. The snapshots are organized in a snapshot matrix (whose columns are the snapshots), as

$$\mathbf{V}_1^K = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K]. \quad (15)$$

In the sequel, the accuracy of the various approximations of the snapshots will be measured in terms of the *relative root mean square* (RMS) error, defined as

$$\text{RRMS error} = \sqrt{\frac{\sum_{k=1}^K \|\mathbf{v}_k^{\text{approx}} - \mathbf{v}_k\|_2^2}{\sum_{k=1}^K \|\mathbf{v}_k\|_2^2}}, \quad (16)$$

where $\|\cdot\|_2$ denotes the usual Euclidean norm. The right hand side of (16) measures the collective RMS error in space and time normalized with the spatiotemporal RMS norm of the snapshots. Obviously, other norms (omitted here for the sake of brevity) would be more appropriate when the dynamics exhibit disparate localized behavior in space and/or time.

For simplicity, only the temporal expansion (5) will be considered in this section. The spatial expansion (9) is obtained by just interchanging the roles of x and t .

2.1. The HODMD Method. The HODMD algorithm proceeds in two steps (see [8] for more details).

Step 1 (dimension reduction). Applying truncated *singular value decomposition* (SVD) [38] to the snapshot matrix \mathbf{V}_1^K yields

$$\mathbf{V}_1^K \simeq \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^\top \equiv \mathbf{P}\widehat{\mathbf{T}}_1^K \quad (\text{or } \mathbf{v}_k = \mathbf{P}\widehat{\mathbf{t}}_k), \quad (17)$$

$$\text{with } \widehat{\mathbf{T}}_1^K = \mathbf{\Sigma}\mathbf{Q}^\top,$$

where the matrix $\widehat{\mathbf{T}}_1^K$ is known as the dimension-reduced snapshot matrix and their columns, the *dimension-reduced snapshots*, $\widehat{\mathbf{t}}_k$. The number of retained modes, which is the spatial complexity defined above, M , is determined by a (tunable) tolerance ε_1 , requiring that the RRMS error of the approximation, as defined in (16), be smaller than ε_1 ; this error is easily computed in terms of the singular values using well-known SVD formulae [38]. For noisy data we may choose ε_1 comparable to the noise level, which helps to clean the data [39]. Now we note that if we had a DMD expansion for the dimension-reduced snapshots, as

$$\widehat{\mathbf{t}}_k \simeq \sum_{n=1}^N a_n \widehat{\mathbf{u}}_n e^{(\delta_n + i\omega_n)t_k} \quad \text{for } k = 1, \dots, K, \quad (18)$$

which in fact will be computed in the next step, substituting this expansion into (17) (namely, premultiplying (18) by the matrix \mathbf{P}), then we obtain the DMD expansion (5) we are looking for.

Step 2 (the DMD-d algorithm for the dimension-reduced snapshots). Now, we must treat the standard DMD and the HODMD algorithms separately. For ordinary DMD, we consider the Koopman assumption (6) for the dimension-reduced snapshots, which reads

$$\widehat{\mathbf{t}}_{k+1} \simeq \widehat{\mathbf{R}}^t \widehat{\mathbf{t}}_k, \quad (19)$$

where the $M \times M$ matrix $\widehat{\mathbf{R}}^t$ (reduced *Koopman matrix*) is computed from the dimension-reduced snapshots via the pseudoinverse. Now, the reduced expansion (18) is computed as follows. The reduced modes $\widehat{\mathbf{u}}_n$ are normalized eigenvectors of $\widehat{\mathbf{R}}^t$, the exponents are related to the eigenvalues, μ_n , as $\delta_n + i\omega_n = \log \mu_n / \Delta t$, and the amplitudes a_n are computed via least squares fitting (as in optimized DMD [19]) of the dimension-reduced snapshots in expansion (18). This completes the algorithm for standard DMD.

Concerning HODMD, the counterpart of (6) for the reduced snapshots,

$$\widehat{\mathbf{t}}_{k+d} \simeq \widehat{\mathbf{R}}_1 \widehat{\mathbf{t}}_{k+d-1} + \widehat{\mathbf{R}}_2 \widehat{\mathbf{t}}_{k+d-2} + \dots + \widehat{\mathbf{R}}_d \widehat{\mathbf{t}}_k, \quad (20)$$

is treated using the *enlarged snapshots*

$$\widetilde{\mathbf{t}}_k \equiv \begin{bmatrix} \widehat{\mathbf{t}}_k \\ \widehat{\mathbf{t}}_{k+1} \\ \dots \\ \widehat{\mathbf{t}}_{k+d-2} \\ \widehat{\mathbf{t}}_{k+d-1} \end{bmatrix}. \quad (21)$$

Applying Step 1 above (a second dimension reduction that eliminates redundancies between the snapshots and delayed snapshots) gives a new set of enlarged dimension-reduced snapshots, $\widetilde{\mathbf{t}}_k$, which are assumed to satisfy the counterpart of (19), namely, $\widetilde{\mathbf{t}}_{k+1} \simeq \widetilde{\mathbf{R}}^t \widetilde{\mathbf{t}}_k$, where the matrix $\widetilde{\mathbf{R}}^t$ is computed via the pseudo inverse, as above. The eigenvalues of eigenvectors of $\widetilde{\mathbf{R}}^t$ permit obtaining the counterpart of (18) for the enlarged dimension-reduced snapshots. Considering only the first M components in this expansion, we obtain the dimension-reduced expansion (18), where the mode amplitudes a_n are computed via least square fitting with the dimension-reduced snapshots, which completes the derivation of (18) in the HODMD method.

Once expansion (18) has been computed either for standard DMD or for HODMD, the expansion is sorted in decreasing order of the mode amplitudes and is further truncated by eliminating those modes such that

$$\frac{a_n}{a_1} < \varepsilon_2, \quad (22)$$

for some tunable parameter ε_2 .

Some remarks about the method are now in order:

- (i) A solver for the DMD- d algorithm described above can be found in [40].
- (ii) It is interesting to note that the enlarged snapshots, containing also the delayed snapshots, are defined for the reduced snapshots (whose size is M), not for the (usually much larger) original snapshots, which, as anticipated, would greatly increase the computational cost of the method. In fact, the ratio of the computational cost of HODMD to that of standard DMD scales with a first or second power (depending on the comparative values of J , Md , and K) of $J/(Md)$, which is of order one in most applications (but obviously not always).
- (iii) Considering periodic or quasi-periodic dynamics, and measuring the accuracy of the DMD reconstruction with the RRMS error defined in (16), the method is quite robust in connection with the tunable parameters d , ε_1 , and ε_2 . In particular, the plot of the RRMS error versus d is fairly flat near the minimum, which means that the selection of d is not critical. In principle, if the data were exact, the smaller ε_1 and ε_2 , the better the approximation. However, in the presence of noise or discretization/truncation errors, ε_1 and ε_2 must be not too small to avoid that the method captures unphysical modes. In other words, these two parameters must be comparable to the noise/errors level, but see the next remark. Finally, the timespan $t_K - t_1$ and the time shift between snapshots, Δt , must be selected as somewhat larger and smaller than the largest and smallest involved periods, respectively.
- (iv) An obvious trade-off of the HODMD method is that we have an additional tunable parameter d , which

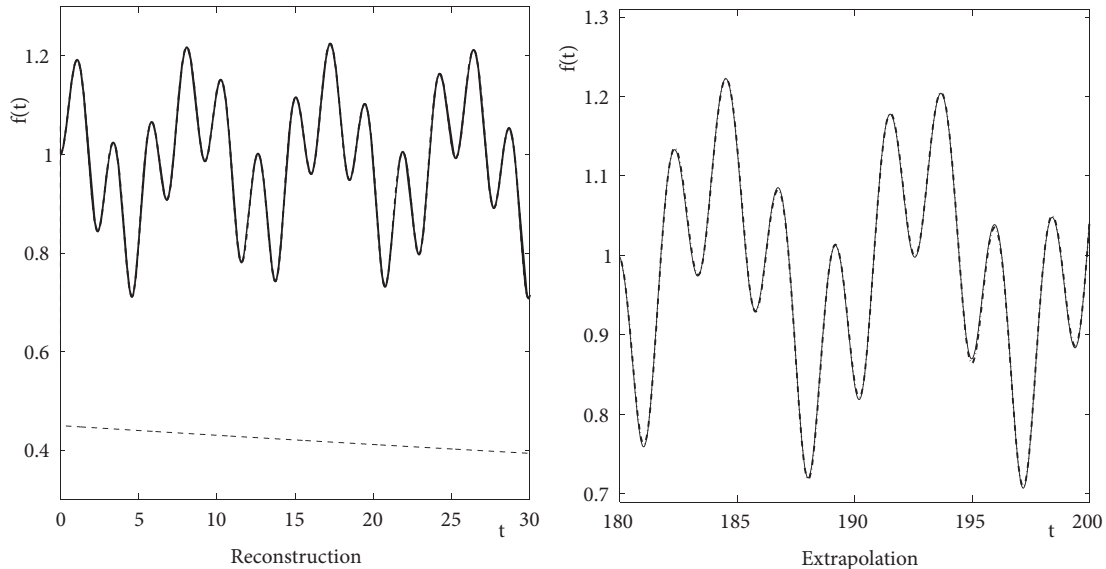


FIGURE 2: Toy model function defined in (16) (thin solid lines) and the reconstruction using DMD-800 (thick dashed lines) and DMD-1 (thin dashed line).

is not present in standard DMD and must be chosen via trial-and-error. However, as explained above, the method is robust in connection with the parameter d .

- (v) The parameters ε_1 and ε_2 can be seen as governing spatial and temporal truncation, respectively. For finite accuracy and given values of the tunable parameters, a new application of the method to the HODMD reconstructed solution does not necessarily give the same reconstruction. This is specially interesting for noisy data, in which the iterative application of the method (*iterative HODMD*, see [41]) improves noise filtering (beyond the well-known [39] noise filtering effect associated with the application of truncated SVD in Step 1). It is to be noted that iterating HODMD with fixed tolerances ε_1 and ε_2 eliminates both spurious SVD modes and spurious DMD frequencies, decreasing along the iterations both the spatial and spectral complexities, M and N , respectively. The iteration is terminated when M and N do not decrease further. By choosing ε_1 and ε_2 comparable to the noise level, the method permits uncovering physical phenomena that were masked by noise in the original data. A striking example of the ability of the iterative HODMD method to filter noise and uncovering the physically relevant dynamics will be given in Section 3.2.
- (vi) When more than one spatial coordinate is present, and spatial discretization is made in a structured mesh in some of the spatial coordinates, the given data can be organized in a *snapshot tensor* instead of a snapshot matrix. In this case, standard DMD can be substituted in Step 1 (see [41]) by HOSVD [42], a well-known extension of standard DMD by Tucker [43]. HOSVD decomposes the tensor as a linear

combination of tensor products of modes in the various directions. The DMD- d algorithm is applied separately (perhaps with different values of d) to the various sets of modes.

2.2. *Illustration of the HODMD Method in a Toy Model.* Let us first illustrate the method in the following quasi-periodic simple toy model

$$v(t) = \sqrt{1 + \frac{1}{2} \sin t \sin(\sqrt{3}t)}. \quad (23)$$

This model exhibits the incommensurable fundamental frequencies $\sqrt{3} \pm 1$, which are somewhat disparate from each other and their joint harmonics. The spatial complexity is $M = 1$, while the spectral complexity N (which increases as the accuracy of the reconstruction is required to be more and more strict) is larger than one. The convenience of using delayed snapshots when the spatial complexity is 1 was already mentioned in [7]. The HODMD algorithm DMD-800 is applied to a set of $K = 2000$ snapshots in the timespan $0 \leq t \leq 32$, with tolerances $\varepsilon_1 = 10^{-10}$ and $\varepsilon_2 = 10^{-4}$. The method identifies $N = 27$ frequencies and reconstructs the snapshots with RRMS error, as defined in (16), $\sim 2 \cdot 10^{-4}$ in this interval. Thus, the original quasi-periodic function and its reconstruction are plot-indistinguishable (see Figure 2-left). Moreover, the extrapolation of this reconstruction to the interval $180 \leq t \leq 200$ (Figure 2-right) is still quite good, since the RRMS error is $\sim 2.7 \cdot 10^{-3}$. On the contrary, the standard DMD algorithm DMD-1 (Figure 2-left, thin dashed line) identifies just one spurious frequency and gives a $O(1)$ RRMS error.

More realistic applications of the HODMD method and its iterative extension are considered in the next section.

3. Applications of HODMD

The discussion about spectral and spatial complexities is a key point to understand the wide range of applications of HODMD. As anticipated in Section 1 and further explained in [11], there are several natural cases in which the spectral complexity is larger than the spatial complexity. Here, we shall concentrate on two such cases, one driven by numerical data from the complex Ginzburg-Landau equation (CGLE) and another by experimental data from the zero-net-mass-flux (ZNMF) jet, which will be considered in Sections 3.1 and 3.2, respectively. Additional applications will be briefly addressed in Section 3.3.

3.1. Application to the CGLE. Let us consider the following CGLE with periodic boundary conditions:

$$\partial_t A = (1 + i\alpha) \partial_{xx}^2 A + \mu u - (1 + i\beta) |A|^2 A, \quad (24)$$

$$A(-1, t) = A(1, t), \quad (25)$$

$$\partial_x A(-1, t) = \partial_x A(1, t).$$

The dependent variable A is complex and can be seen as the *complex amplitude* when using (24) as the weakly nonlinear “normal form”, which applies at the onset of oscillatory instabilities in many relevant physical systems [34, 35]. The CGLE is a well-known paradigm of pattern forming systems that exhibits complex dynamics [44] due to the modulational instability if $\alpha\beta < -1$ (Newell’s condition [44]) and μ exceeds a threshold value. For typical values of α , β , and μ , the CGLE possesses an inertial manifold contained in a linear subspace of dimension 20-30 (upper limit of the spatial complexity), while the dynamics may be complex, meaning that the spectral complexity can be large. Thus, the standard DMD-1 algorithm fails except for the simplest monochromatic attractors [8].

Problem (24)-(25) is invariant under the $O(2) \times SO(2)$ symmetry group generated by spatial translations and reflections, and phase shifts in A ,

$$\begin{aligned} x &\longrightarrow x + c_1, \\ x &\longrightarrow -x, \\ A &\longrightarrow A e^{ic_2}. \end{aligned} \quad (26)$$

Because of the latter symmetry, it is convenient to write down expansion (1) for the CGLE as

$$A(x, t) = e^{i\omega_0 t} \sum_{n=1}^N a_n u_n(x) e^{i\omega_n t}, \quad (27)$$

where the frequencies $\omega_1, \omega_2, \dots$ are linear combinations with integer coefficients of some fundamental frequencies, which are generally incommensurable with ω_0 . Thus, the dynamics of A are more complex than those of $|A|$. In other words, A is periodic if $|A|$ is stationary, and A represents a 2-torus if $|A|$ is periodic. In the sequel, we consider just two attractors, one

such that $|A|$ is periodic and another such that $|A|$ is quasi-periodic. Both attractors are obtained for

$$\begin{aligned} \alpha &= -10, \\ \beta &= 10, \end{aligned} \quad (28)$$

and two values of μ . Note that since $|\alpha|$ and $|\beta|$ are both somewhat large, dispersion and nonlinear detuning are large compared to diffusion and nonlinear damping, respectively, which is demanding from the computational point of view and promotes complex dynamics. For the construction of the databases, the equation is solved numerically using a Crank-Nicolson plus Adams-Bashforth scheme [45] for temporal discretization, with time step $\Delta t = 10^{-5}$, while the second-order spatial derivative is discretized using centered finite differences in a uniform grid of 1000 points. HODMD is applied to a set of 2000 snapshots, collected in an appropriate time interval of length 1, disregarding a transient time-stage. In the two cases considered below, a good value of the index d when applying the HODMD algorithm is $d = 50$, while the remaining tunable parameters are $\varepsilon_1 = 10^{-8}$ and two values of ε_2 , namely,

$$\begin{aligned} \varepsilon_2 &= 5 \cdot 10^{-3} \text{ (high accuracy)}, \\ \varepsilon_2 &= 10^{-4} \text{ (low accuracy)}. \end{aligned} \quad (29)$$

Selecting these values of the tunable parameters as common for the two cases considered below illustrates the robustness of the method, which is further checked noting that low and high accuracy computations are consistent among each other, and the results coincide when different timespans are used in the computations.

The first case to be considered is that for $\mu = 20$, in which using the DMD-50 algorithm, the RRMS reconstruction error (see (16)) is $\sim 6 \cdot 10^{-3}$ and $\sim 2 \cdot 10^{-5}$ for low and high accuracy, respectively, retaining 6 and 12 DMD modes, respectively. The semilogarithmic plot of the DMD amplitude versus frequency is plotted in Figure 3-left, where it can be seen that the relevant points are approximately contained in two straight lines centered at $\omega_0 = 11.6$, which means that expansion (27) converges spectrally; using the standard DMD method (DMD-1 algorithm), instead, gives spurious results, plotted with red symbols in the figure. The remaining points obtained via the DMD-50 algorithm lie equispaced in frequency, with a distance between them equal to $\omega = 118.7$, which means that the frequencies ω_n appearing in expansion (27) are all (positive or negative) harmonics of ω . Thus, $|A|$ is time-periodic (see Figure 3, middle), with a period $2\pi/\omega$, while A is quasi-periodic (2-torus), with fundamental frequencies ω_0 and ω . Since ω_0 is small compared to ω the pattern is slowly modulated (Figure 3, right). The spatial modes appearing in the expansion (27), u_n , which are not plotted for the sake of brevity, exhibit a reflection symmetry around $x = -0.17$ (indicated with a vertical white line in Figure 3), meaning that the whole pattern also exhibits this symmetry, consistently with the invariance of the problem under the actions (26). This implies that the pattern can be seen as a standing wave (SW), as indicated with the vertical

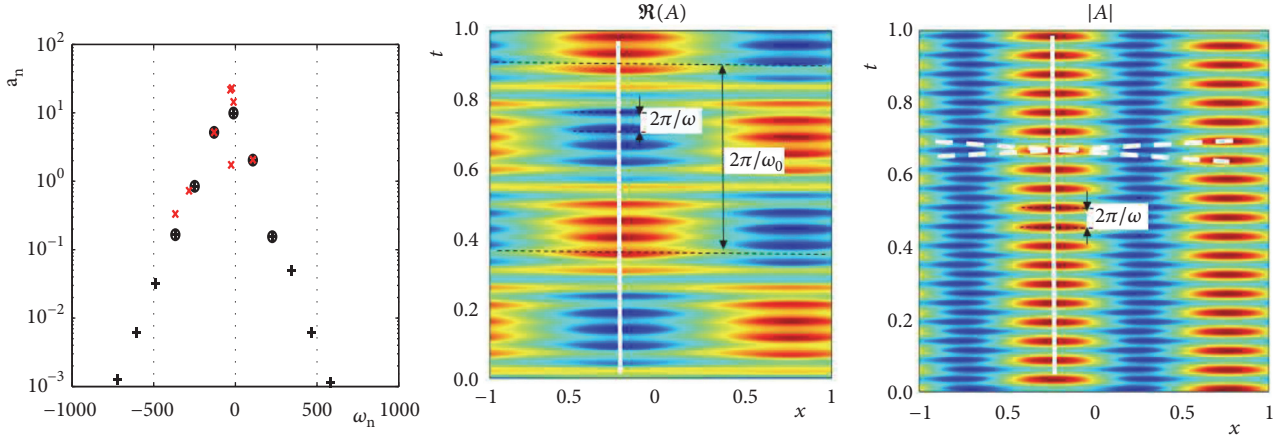


FIGURE 3: The CGLE for $\mu = 20$, plotting the amplitudes versus frequencies in the DMD expansion (left plot) for low (black circles) and high (black crosses) accuracy computations using DMD-50, and using DMD-1 (red symbols), and the color map of $\Re(A)$ (middle plot) and $|A|$ (right plot).

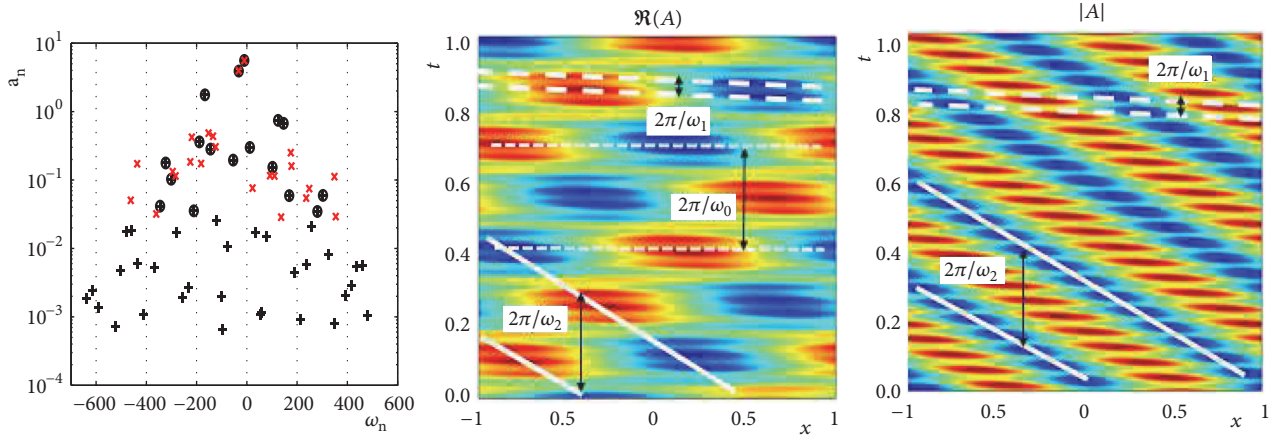


FIGURE 4: Counterpart of Figure 3 for $\mu = 22$.

while solid line in Figure 3, but it can also be considered as either a right or left traveling wave (TW), as indicated with the oblique dashed lines in this figure. This dual interpretation cannot be ascertained from the shape of the modes u_n , but this issue will be revisited and further checked in Section 5.1.

As a second case, we consider the value $\mu = 22$ (Figure 4), in which the dynamics are more complex. Again, the DMD-1 algorithm yields spurious results. Instead, for low and high accuracy (see (29)), the DMD-50 algorithm retains 17 and 49 modes, giving a RRMS reconstruction error (see (16)) $\sim 8 \cdot 10^{-3}$ and $\sim 2 \cdot 10^{-4}$, respectively. The main difference with the case considered in Figure 3 is that now the plot of the amplitude versus the frequency forms a triangular pattern that also decays spectrally but shows that, in addition to $\omega_0 = -21.2$, the frequencies in the expansion in the right hand side of (27) are linear combinations with integer coefficients of the fundamental frequencies $\omega_1 = 156.6$ and $\omega_2 = 22.2$, which are not easily identified in Figure 4, left. Thus, guessing that the three fundamental frequencies are incommensurable, the attractor for $|A|$ is contained in a 2-torus while that for A lies in a 3-torus. Finally, the middle and right plots in Figure 4

show that the pattern is a TW but, as in the former case, elucidating this issue from the shape of the modes $u_n(x)$ appearing in (27) turns out to be a difficult (and unnatural) issue. The STKD method, which will be applied to the CGLE in Section 5.1, instead, will solve this issue in a natural way.

In summary, the HODMD method is able to precisely compute the temporal DMD expansion for the CGLE, giving robust results and distinguishing between periodic and quasi-periodic phenomena. However, identifying SWs and TWs requires additional ingredients, which will be included in the STKD method, developed in Section 4 and applied to the CGLE in Section 5.1.

3.2. Application to the Zero-Net-Mass-Flux Jet. A zero-net-mass-flux (ZNMF) jet [46, 47] is a pulsatile jet promoted by a piston or membrane that interchanges fluid flow through an orifice with an outer ambient fluid that is at rest far from the orifice. The net mass flux is zero, but at large Reynolds number there is a nonzero (and large) net flux of momentum that, in addition to its own scientific interest, is useful in various fields. For instance, synthetic jet actuators promote pulsatile

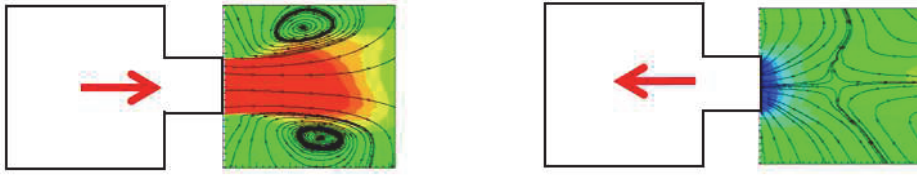


FIGURE 5: Pseudo-streamlines in a meridional section for two representative snapshots of the injection (left) and suction (right) phases in the ZNMF jet.

jets [48] that are useful in, e.g., mixing enhancement [49], jet vectoring [50], heat transfer [51], and active flow control of boundary layer separation [52]. These jets are also relevant in nature, in connection with squid, octopus, salp, and jellyfish swimming [53].

It must be noted that the nonzero momentum flux at large Reynolds number is due to the fact that, even if the periodic forcing is temporally reflection symmetric, the flow is highly nonreflection symmetric, as illustrated in Figure 5, where an approximately rotationally symmetric jet produced by a circular orifice is considered. As can be seen, a vortex ring is formed in the injection phase that travels downstream, while the suction phase is characterized by a saddle point that separates the fluid reentering the cavity from that still moving downstream.

The ZNMF jet promotes a complex flow structure that involves various, disparate spatiotemporal scales, whose understanding remains as an open topic. Thus, this is a convenient problem to test the performance of the HODMD method. Also, in order to test the noise filtering ability of the iterative HODMD method, the analysis will be based on experimental data resulting from particle image velocimetry (PIV), as obtained [46, 47] in the experimental facility for ZNMF jets of the Laboratory for Turbulence Research and Combustion at Monash University. Such measurements exhibit an error $\sim 2.4\%$ at the 95% confidence level. Two nondimensional parameters characterize the flow conditions, namely the *Reynolds and Strouhal numbers*, defined as $Re = UD/\nu$ and $St = fD/U$, respectively, where D is the jet orifice diameter, ν is the kinematic viscosity, U is related to the peak oscillation velocity of the piston \widehat{V}_p as $U = 5\widehat{V}_p/\sqrt{2}$ (see [41] for details), and f is the piston oscillation frequency. In the present case, we have $Re = 13329$ and $St = 0.03$, which give a flow lying in the laminar-transitional regime. The flow is fully turbulent in the far field, but laminar in the near field (near the orifice), which will be the region of the jet considered here; see [41] for further details on the experimental facility and the PIV measurement.

The orifice is circular and the resulting near field flow is approximately axi-symmetric, at least in the large scale. The experimental data consists in $K = 1872$ snapshots representing 12 piston oscillation cycles, giving the streamwise and radial velocity components, $u(x_i, r_j, t_k)$ and $v(x_i, r_j, t_k)$, respectively, at $I \times J = 52 \times 68$ grid points in a meridional rectangle (interrogation window), where x and r are the streamwise and radial coordinates, respectively, with origin at the center of the orifice and t is the time variable, with

origin at the beginning of the suction phase. The time variable has been rescaled such that the (nondimensional) oscillating frequencies are measured in terms of the associated Strouhal number defined above, meaning that the piston oscillating period is $T = 2\pi/0.03$. Thus, we have a snapshot tensor (instead of a matrix) and HOSVD must be used instead of standard SVD in the first step of the HODMD method (see [41] for details). Assuming permanent dynamics (zero or very small growth rates) the aim of the iterative HODMD method is to obtain an expansion of the form

$$\mathbf{v}(x, r, t) \approx \sum_{n=1}^N \mathbf{u}_n(x, r) e^{i\omega_n t}, \quad (30)$$

The iterative DMD- d algorithm has been applied to obtain this expansion using two values of the index d , namely, $d = 1$ (standard DMD) and $d = 700$ (strictly higher order DMD), with two tolerance sets, namely,

$$\begin{aligned} \varepsilon_1 = \varepsilon_2 &= 2.4 \cdot 10^{-2} \text{ (low accuracy),} \\ \varepsilon_1 = \varepsilon_2 &= 10^{-2} \text{ (high accuracy),} \end{aligned} \quad (31)$$

which correspond to the estimated experimental error and more precise computations, respectively. Using the latter tolerances, below the experimental error, will permit elucidating the ability of the iterative HODMD method to uncover physical phenomena that are masked by the experimental errors in the given data.

Figure 6 shows the amplitude versus frequency (nondimensionalized as the Strouhal number) obtained in the analyses. DMD-1 captures 11 modes, all of them spurious.

On the contrary, DMD-700 captures $N = 19$ and $N = 49$ DMD modes using low and high accuracy, respectively. It is remarkable that the spatial complexity M is much smaller than the spectral complexity, N , since $M = 5$ and $M = 11$ for low and high accuracy, respectively, which justifies the failure of DMD-1. The method shows that the flow is periodic in the near field, with dominant frequency $St = 0.03$ (piston oscillation frequency). Since the data are real, the DMD modes appear in complex-conjugate pairs, meaning that the retained modes contain the mean flow ($St = 0$) and 9 and 24 nonzero harmonics for low and high accuracy, respectively.

The suction phase (see Figure 5-left) is not quite demanding, since it just shows a saddle point at the axis of the jet. The reconstructions using DMD-700 with low and high accuracies are qualitatively similar to their counterparts for

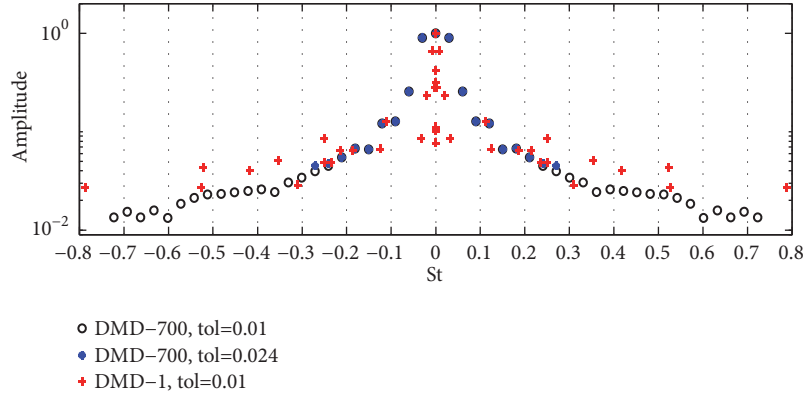


FIGURE 6: Amplitude versus St obtained for low accuracy via DMD-1 (red crosses) and DMD-700 (blue circles) and for high accuracy using DMD-700 (black circles).

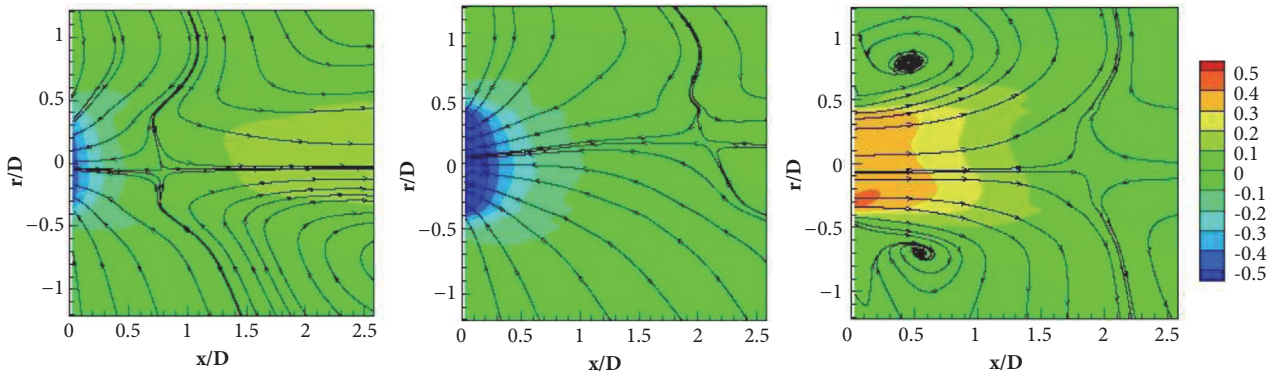


FIGURE 7: Two snapshots in the suction phase, for $t = T/18$ (left plot) and $t = 2T/9$ (middle plot), and a snapshot at the very beginning of the injection phase, for $t = T/2 + T/72$ (right plot).

the original PIV data. The main difference is that the iterative HODMD method gives smoother results with high accuracy than with low accuracy, and both are smoother than the original PIV data. This is an indication of the ability of the method to filter experimental errors, but this issue cannot be confirmed because the actual physical data are not available. Concentrating on the reconstructions with low accuracy, it is seen that the saddle point is created near the orifice at the beginning of this phase and travels downstream to a distance that is approximately equal to $2D$, where it remains steady for most part of the suction phase (Figure 7, left and middle plots). This saddle point is blown downstream at the beginning of the injection phase, where a new saddle point is created near the orifice that travels downstream very fast (Figure 7, right plot) and leaves the near field, in a short timespan of the order of $T/100$. This saddle point must be present for topological reasons, but it had never been seen in studies previous to [41], just because it is quite short living. In fact, it is not present in any of the snapshots used to apply the HODMD method. It must be noted that this saddle point would be present (with some spatial noise) in the experimental snapshots if the sampling frequency was sufficiently large. Thus, the identification of this saddle point illustrates well the advantages of the temporal interpolation that is implicit in the HODMD method.

The injection phase (see Figure 5, right) is more interesting in the present context. Figure 8 shows a comparison of the original data with the reconstructions provided by DMD-700 for low and high accuracy. As can be seen, the reconstructions are much smoother than the original PIV data, as it happened in the suction phase. However, the reconstruction with high accuracy (recall that this is below the experimental error) is qualitatively different, since it shows some small vortices leaving the edge of the cavity that cannot be even guessed in the original PIV data and do not appear in the low accuracy reconstruction because they involve a quite small energy. These small vortices are smooth, which may indicate that they are not artifacts associated with higher order modes resulting from experimental errors. However, this cannot be ascertained since the actual physical dynamics is not available. Therefore, we have conducted some numerical simulations to elucidate this point. The value of the Reynolds number in the experimental data considered above, $Re \sim 13000$, is a quite large value that would need huge numerical resources, even using the computationally efficient solver Nek5000 [54] (an open source spectral element code). Therefore, we have performed computations using the lower Reynolds number $Re=1000$ and the same Strouhal number as in the experimental data considered above, namely, $St=0.03$. The numerical counterpart of the snapshot considered in

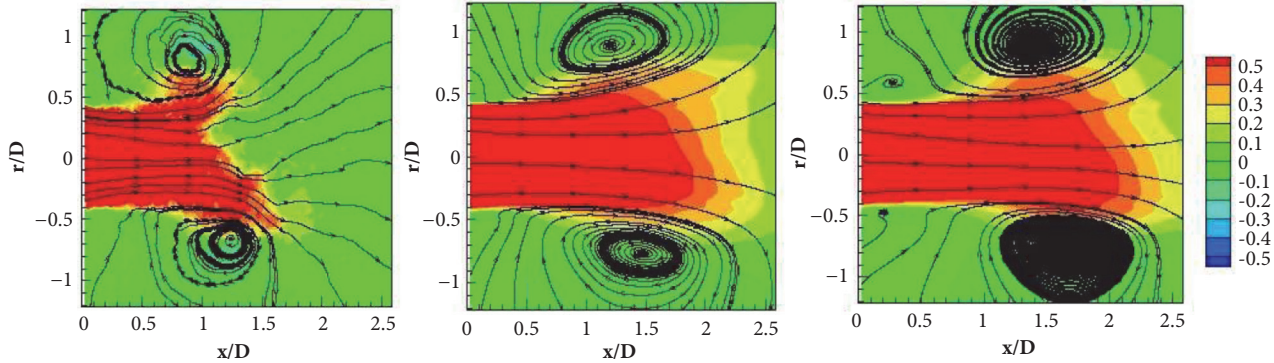


FIGURE 8: Instantaneous pseudo-streamlines and streamwise velocity contours for a representative snapshot ($t = T/9$) in the injection phase, showing the original PIV data (left plot) and the reconstructions using DMD-700 for low (middle plot) and high (right plot) accuracy.

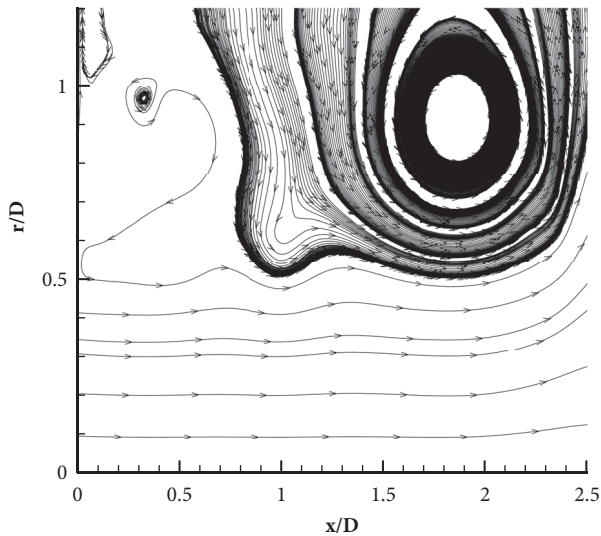


FIGURE 9: Counterpart of the snapshot considered in Figure 5 but as numerically computed for $Re=1000$.

Figure 8 is given in Figure 9, which confirms the presence of the small vortices.

In summary, the results above illustrate well the ability of the iterative HODMD method to both (i) take advantage of temporal interpolation to uncover fast events (such as the saddle point at the beginning of the injection phase) that are not present in the snapshots and (ii) filter experimental errors so well that some small patterns (such as the small vortices in the injection phase) can be identified that were completely masked by the experimental noise. In addition, the analysis above permits constructing a purely data-driven (from experimental data) ROM that reproduces well the periodic behavior of the ZNMF jet and shows a fast online operation.

3.3. Additional Applications. Some additional applications are now briefly quoted.

The HODMD method permits identifying flow instabilities by monitoring those DMD modes whose growth rate

changes sign. On the other hand, as anticipated, retaining only those terms appearing in the DMD expansion (1) that exhibit zero (or very small) growth rate, the final attractor may be obtained from transient data. This latter idea can be used to strongly decrease the CPU time required to compute the final attractor, avoiding continuing the numerical integration until final attractor is reached, which may require an extremely large timespan, specially near bifurcation points. Transient dynamics are usually found in the initial stage of a numerical simulation, where a large number of modes (either physical or spurious) appear simultaneously.

Let us illustrate these two applications by considering the cylinder wake, namely, the wake behind a circular cylinder immersed in a free stream with constant, spatially uniform velocity, which is a basic fluid dynamics problem [55, 56]. This problem depends on a unique nondimensional parameter, the Reynolds number $Re = UD/\nu$, where U is the free stream velocity, D is the cylinder diameter, and ν is the dynamic viscosity of the fluid, assumed to be incompressible. Thus, the nondimensional velocity and pressure obey the continuity and Navier-Stokes equations:

$$\nabla \cdot \mathbf{v} = 0, \quad (32)$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \frac{1}{Re} \Delta \mathbf{v}, \quad (33)$$

which are usually integrated imposing \mathbf{v} is equal to the free stream velocity at the inlet boundary, no-slip boundary conditions at the boundary of the body, appropriate (non-reflecting) boundary conditions at the sides and exit of the computational domain, and periodic boundary conditions in the spanwise direction, with a period L_z . The wake is two-dimensional and steady for small Re , but at $Re \approx 46$ it exhibits a first unsteady instability (Hopf bifurcation [57, 58]) that produces a still two-dimensional but unsteady, periodic von Karman vortex street flow. The flow remains orbitally stable up to $Re \approx 190$, where it suffers a secondary bifurcation (Floquet multiplier = 1) and becomes three-dimensional [59] for spanwise period $L_z > 3D$ (with D the cylinder diameter). The resulting mode is a long-wave, periodic, flow known as *mode A*. For larger values of Re , at $Re=259$, a second periodic short-wave mode appears if $L_z > 1.5D$ that is known as *mode*

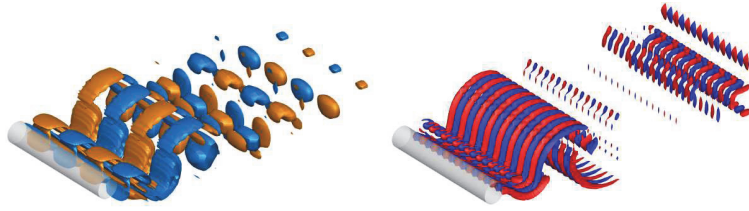


FIGURE 10: Streamwise velocity contours for modes A (left) and B (right) obtained at $Re= 280$, with and $L_z = 6.99$.

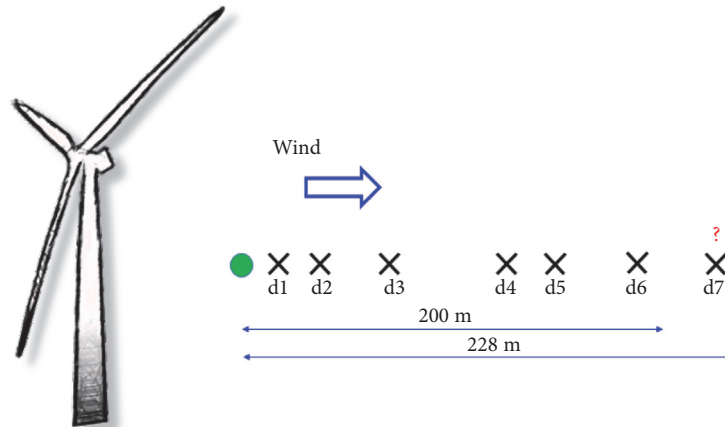


FIGURE 11: Measurements upstream the wind turbine in the LiDAR experimental campaign. Data are available in the six planes that are closer to the wind turbine, while the data on the seventh plane are to be predicted.

B. Modes A and B are illustrated in Figure 10. These modes are usually obtained via Floquet stability analysis of the basic two-dimensional periodic solution [59–61], but they can also be obtained via HODMD applied to the whole dynamics obtained via numerical simulations [62]. It is relevant to note that for higher values of the Reynolds number, at $Re \approx 380$, a new instability takes place that is associated with a pair of complex-conjugate Floquet multipliers that give a family of *quasi-periodic solutions* [61]. These modes were also identified in [62].

As for temporal extrapolation via the HODMD method, this idea has been used [11] precisely in the three-dimensional cylinder wake, with $Re=220$ and $L_z = 4$. The numerical data was obtained using the above-mentioned numerical solver Nek5000 [54]. Temporal HODMD extrapolation divided by five the CPU time required computing the final attractor with a reasonable accuracy; see [11] for further details.

Additional applications involving numerically generated data include the analysis of the turbulent wake of a cross flow wind turbine [63] and an off-shore wind turbine [64], as well as the analysis of wakes interaction [65].

Concerning analyses based on experimental data, a good example, linked to the field of renewable energies, is the postprocess of light detection and ranging (LiDAR) [66, 67] measurements. LiDAR is an experimental method for the remote measurement of the line-of-sight component of wind speed, usually employed in the wind energy industry with different goals, including the maximization of the power generated by the wind turbine. Measurements are made at

a few (say, M) points upstream the wind turbine, based on detection of the Doppler shift for light backscattered from natural aerosols transported by the wind in the atmosphere. LiDAR measurements offer M time-dependent signals that present high noise levels (15 – 20%). Thus, the spatial complexity M is small, making HODMD a suitable tool to analyze the data and detect the dominant frequencies. A great difficulty is that LiDAR experiments only measure at distances between 20 m and 200 m from the measurement device, which is a restriction in certain situations. Figure 11 shows six planes of LiDAR measurements located at distances between 33 m and 201 m upstream the wind turbine. HODMD is used to analyze such data and to predict the wind velocity at a seventh distance (228 m), which is well beyond the LiDAR device measurement range. The idea used in [68] is to extrapolate the spatial modes in the DMD expansion, guessing the data at the seventh distance with relative errors $\sim 2\%$. The computational cost for predicting these measurements is negligible, making it possible to easily update the model in real-time (leading to a fast, purely data-driven ROM).

Additional applications using experimental data where the HODMD method have proved to give good results including nonisothermal flows around square objects in a channel [69] and the analysis of aeroelastic data in aircraft flight tests [70]. The aim in the latter application is to obtain the natural aeroelastic frequencies, and the associated damping rates, of the various aircraft parts (e.g., wings or fuselage) from data obtained at some sensors along the

aircraft when this is subject to various types of impulsive excitation.

4. Spatiotemporal Koopman Decomposition

As seen in the previous sections, HODMD is suited for analyzing complex dynamics and provides accurate signal descriptions in terms of dominant frequencies and DMD modes. Nevertheless, the spatial components of the data analyzed may bring some relevant information that, combined with the temporal information, helps in the analysis and interpretation of the data. The main benefit of this idea, which gives rise to the spatiotemporal Koopman decomposition (STKD) method presented below, is that it further simplifies the spatiotemporal description of complex data in terms of TWs. The STKD method leads to data approximations of the type of (11), which for the sake of clarity is rewritten here as

$$\mathbf{v}(x, \mathbf{y}, t) = \sum_{m,n} a_{mn} \mathbf{u}_{mn}(\mathbf{y}) e^{(\nu_m + i\kappa_m)x + (\delta_n + i\omega_n)t}, \quad (34)$$

for $x_1 \leq x \leq x_1 + X$ and $t_1 \leq t \leq t_1 + T$. This expansion can be seen as a linear combination of *monochromatic* TWs, with phase velocities $c_{mn} = -\omega_n/\kappa_m$, which may be organized in various types of more complex standing or propagative patterns [28]. For a given finite accuracy, it is frequent that only some combinations of m and n need to be retained that, moreover, are grouped along straight lines in ω versus κ plane (called the ω - κ *diagram* below), which may be either horizontal or oblique and correspond to either *standing* or *traveling* patterns, respectively.

Pure TWs are a particular case, in which the number of spatial and temporal modes coincide, the amplitude matrix a_{nm} is diagonal, $\delta_n = \nu_m = 0$, and $\omega_n = -c\kappa_m$. Then, (34) can be expressed as

$$\mathbf{v}(x - ct, \mathbf{y}) \simeq \sum_{n=1}^N a_{nm} \mathbf{u}_{nm}(\mathbf{y}) e^{i\kappa_m(x-ct)}. \quad (35)$$

In this case, the relevant points in the $\omega - \kappa$ diagram are in a straight line passing through the origin. If $c = 0$, then the pattern is a pure SW. When the points in the $\omega - \kappa$ diagram can be organized in families of horizontal or oblique, parallel straight lines, then the pattern is a modulated SW or TW, respectively. Combinations of these two give more general patterns; see the examples presented below.

In some cases, the dynamics present oscillatory behavior in more than one spatial direction. Concentrating for simplicity in the case in which two spatial directions x and y are involved, the STKD method provides an expansion of the type

$$\mathbf{v}(x, y, z, t) \simeq \sum_{m_1, m_2, n} a_{m_1 m_2 n} \mathbf{u}_{m_1 m_2 n}(z) \cdot e^{(\nu_{m_1}^1 + i\kappa_{m_1}^1)x + (\nu_{m_2}^2 + i\kappa_{m_2}^2)y + (\delta_n + i\omega_n)t}, \quad (36)$$

which also give TWs. For instance, if the temporal and spatial growth rates are all equal to zero and the wavenumbers and frequencies are such that

$$\frac{\omega_n}{\sqrt{(\kappa_{m_1}^1)^2 + (\kappa_{m_2}^2)^2}} = -c, \quad (37)$$

$$\kappa_{m_2}^2 = d\kappa_{m_1}^1,$$

then pattern (36) represents a pure TW with phase velocity c that propagates in the direction of the lines $d\mathbf{y} - x = \text{constant}$. As in the simpler case in which only one spatial direction is involved, pure SWs correspond to the case $c = 0$, and more general patterns are obtained combining these basic patterns. The extension to the case in which the three spatial variables are involved is straight forward.

4.1. The STKD Method. Let us now present the STKD method, considering the discrete counterpart of (34), with the discrete values of x and t defined as $x_j = x_1 + j\Delta x$ and $t_k = t_1 + k\Delta t$, for $j = 1, \dots, J$ and $k = 1, \dots, K$, with $\Delta x = X/J$ and $\Delta t = T/K$; as in the temporal HODMD method, the continuous decomposition is obtained from the discrete decomposition by just allowing x_j and t_k to take continuous values. In addition, for simplicity, we consider a scalar state variable in the strictly one-dimensional case, in which no dependence on the transversal variable \mathbf{y} is present. Namely, we substitute (34) by

$$v(x_j, t_k) = \sum_{m,n} a_{mn} u_{mn} e^{(\nu_m + i\kappa_m)x_j + (\delta_n + i\omega_n)t_k}, \quad (38)$$

where the modes u_{mn} are complex numbers with unit absolute value. The method proceeds in two steps.

Step 1 (dimension reduction). As in HODMD, truncated SVD with a tunable tolerance ε_1 (which determines the number of retained SVD modes, M) is applied to the $J \times K$ snapshot matrix (15), but in the present case the reduced spatial and temporal modes are scaled differently. In other words, (17) is replaced by

$$\mathbf{V}_1^K \simeq \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^\top \equiv (\widehat{\mathbf{X}}_1^J)^\top \widehat{\mathbf{T}}_1^K, \quad (39)$$

where $M \times J$ and $M \times K$ matrices \mathbf{X}_1^J and \mathbf{T}_1^K are defined as

$$\widehat{\mathbf{X}}_1^J = \sqrt{\Sigma} \mathbf{P}^\top, \quad (40)$$

$$\widehat{\mathbf{T}}_1^K = \sqrt{\Sigma} \mathbf{Q}^\top.$$

These matrices are called the *reduced spatial and temporal snapshot matrices*, respectively, and their columns, $\widehat{\mathbf{x}}_j$ and $\widehat{\mathbf{t}}_k$, the *reduced spatial and temporal snapshots*, respectively. Comparison between (40) and (17) shows that while the SVD singular values were used in the temporal HODMD method to scale the reduced temporal snapshots, they are equidistributed now to rescale the reduced spatial and temporal snapshots, which are both scaled with the square root of the singular values.

Step 2 (computation of the STKD expansion). The higher DMD-d algorithm described in Section 2 is applied to the reduced spatial and temporal snapshots, using appropriate indexes d^x and d^t , respectively, which leads to

$$\hat{\mathbf{x}}_j \approx \sum_{m=1}^M \hat{a}_m^x \hat{\mathbf{q}}_m^x e^{(v_m + i\kappa_m)x_j}, \quad \text{for } j = 1, \dots, J, \quad (41)$$

$$\hat{\mathbf{t}}_k \approx \sum_{n=1}^N \hat{a}_n^t \hat{\mathbf{q}}_n^t e^{(\delta_n + i\omega_n)t_k} \quad \text{for } k = 1, \dots, K. \quad (42)$$

The numbers of retained modes, M and N , are the *spatial and temporal complexities*, respectively, and are selected in terms of the spatial and temporal amplitudes, \hat{a}_m^x and \hat{a}_n^t , respectively, as we did in Section 2. Namely, M and N are the smallest values of the indexes m and n such that

$$\begin{aligned} \frac{\hat{a}_{M+1}^x}{\hat{a}_1^x} &< \varepsilon_2, \\ \frac{\hat{a}_{N+1}^t}{\hat{a}_1^t} &< \varepsilon_2. \end{aligned} \quad (43)$$

for some tunable tolerance ε_2 .

Now, substituting (41)-(42) into the columns of the reduced snapshot matrices $\hat{\mathbf{X}}_1^J$ and $\hat{\mathbf{T}}_1^K$ appearing in (39) yields the STKD expansion (38), with

$$\begin{aligned} a_{mn} &= |\hat{q}_{mn}| > 0, \\ q_{mn} &= \frac{\hat{q}_{mn}}{|\hat{q}_{mn}|}, \end{aligned} \quad (44)$$

where

$$\hat{q}_{mn} = \hat{a}_m^x \hat{a}_n^t (\hat{\mathbf{q}}_m^x)^\top \hat{\mathbf{q}}_n^t. \quad (45)$$

Note that the modes q_{mn} are complex numbers exhibiting unit absolute value, as required, and that the resulting expansion (38) exhibits $M \times N$ spatiotemporal modes. However, this number of modes can be decreased by eliminating those modes such that

$$a_{mn} < \varepsilon_2, \quad (46)$$

where the threshold ε_2 is tunable.

The method described above can be called a *parallel method* because the spatial and temporal variables, x and t , are treated in the same way. The extension of the method to the case in which more than one spatial variable is involved (see, e.g., (34) and (36)) is straight forward, except that the snapshot matrix becomes a snapshot tensor and HOSVD needs to be used instead of SVD in Step 1 (as in Section 3.2). For the sake of brevity, we do not describe this extension here; see [28] for a detailed development and illustration of this extension.

Instead of the parallel method described above, a simpler but less consistent *sequential method* may be used to obtain

the discrete STKD expansion (38) as follows. First, the temporal DMD-d algorithm described in Section 2, with an appropriate index d^t , is applied to the snapshot matrix, which leads to

$$v(x_j, t_k) \approx \sum_{n=1}^N a_n^t u_{nj}^t e^{(\delta_n + i\omega_n)t_k}. \quad (47)$$

As a second step, we apply the spatial DMD-d algorithm, with an appropriate index d^x , to the $N \times J$ snapshot matrix \mathbf{W}_1^J whose (n, j) element is $a_n^t u_{nj}^t$, which leads to

$$a_n^t u_{nj}^t \approx \sum_{m=1}^M a_m^x u_{mn}^x e^{(v_m + i\kappa_m)x_j}. \quad (48)$$

Substituting this into (47) and setting

$$\begin{aligned} a_{mn} &= |a_m^x u_{mn}^x|, \\ u_{mn} &= \frac{a_m^x u_{mn}^x}{|a_m^x u_{mn}^x|} \end{aligned} \quad (49)$$

yields the discrete expansion (38), as required. However, as in the parallel method, the number of terms in this expansion is $M \times N$, which can be decreased by ignoring those modes that satisfy (46).

Two remarks about the sequential method are now in order:

- (i) The order in which the variables x and t are treated above can be interchanged, which generally gives slightly different results.
- (ii) As with the parallel method, when more than one spatial variable is present, the snapshot matrix becomes an snapshot tensor and standard SVD must be replaced by HOSVD in the application of the HODMD method.

Because the parallel method is more consistent, specially when treating more than one spatial variable, it will be this method that will be used in the remainder of this section, unless otherwise stated.

As with the HODMD, the STKD method can be used to analyze experimental data. The noisy artifacts that may cover the actual physical data can be cleaned using the proper tolerances and applying the method iteratively (as in the example presented in Section 3.2). Similarly, the STKD method is suitable for analyzing transient dynamics (as in the example shown in Section 3.3).

4.2. Illustration of the STKD Method in a Toy Model. Let us now apply these two methods in the following toy model, already considered in [29], defined as

$$\begin{aligned} u(x, t) &= (0.5 + \sin x) \\ &\cdot [2 \cos(\tilde{\kappa}_1 x - \tilde{\omega}_1 t) + 0.5 \cos(\tilde{\kappa}_2 x - \tilde{\omega}_2 t)], \end{aligned} \quad (50)$$

with $\tilde{\kappa}_1 = 2$, $\tilde{\kappa}_2 = 10$, $\tilde{\omega}_1 = 2\pi/45$, and $\tilde{\omega}_2 = \sqrt{10}$. It is a simple (but demanding because $\tilde{\omega}_1$ and $\tilde{\omega}_2$ are disparate from each

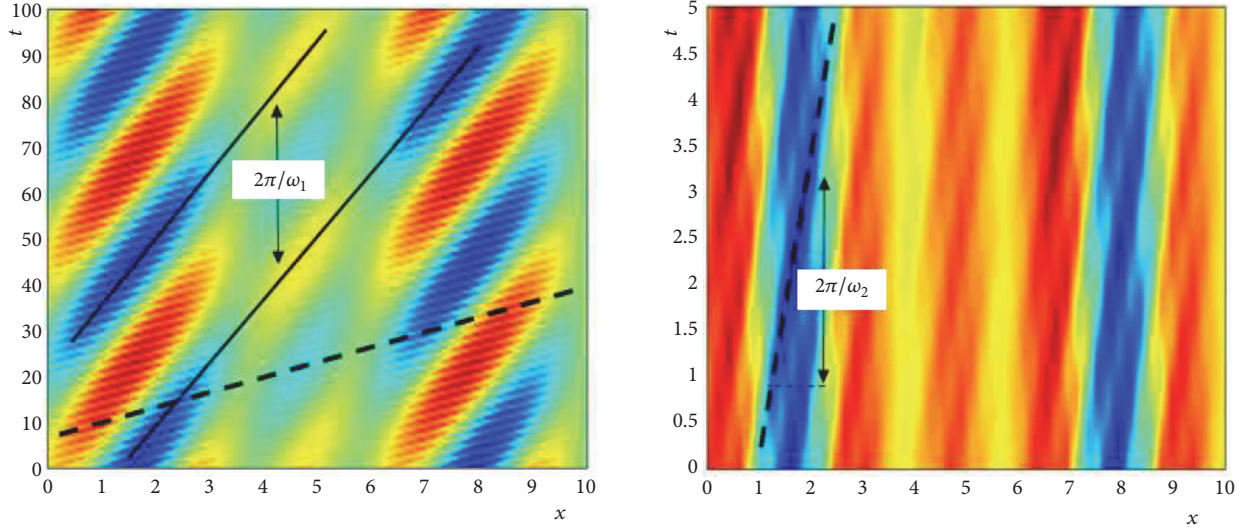


FIGURE 12: Color map of the toy model (50) (left plot) and its restriction to the sampled spatiotemporal square where the STKD method is applied (right plot).

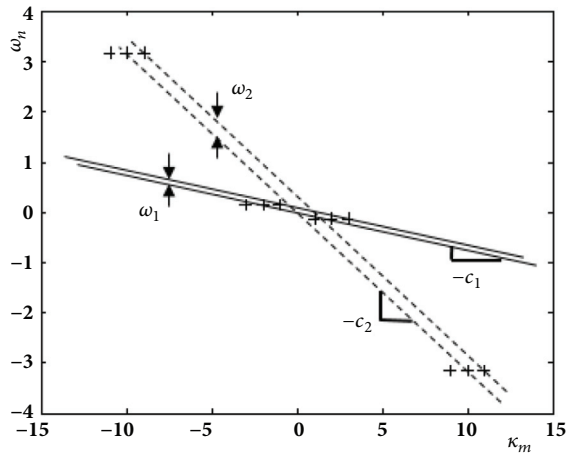


FIGURE 13: $\omega - \kappa$ diagram of the toy model (50).

other; see Figure 12-left) model that can be cast into the form (34), with $v_m = \delta_n = 0$, involving 12 wavenumbers, namely, $\pm\tilde{\kappa}_1, \pm\tilde{\kappa}_2, \pm(\tilde{\kappa}_1 \pm 1), \pm(\tilde{\kappa}_2 \pm 1)$, and 4 frequencies, namely, $\pm\tilde{\omega}_1$ and $\pm\tilde{\omega}_2$. Thus the spectral spatial and temporal complexities are $M = 12$ and $N = 4$, respectively. The pattern is temporally quasi-periodic because $\tilde{\omega}_1$ and $\tilde{\omega}_2$ are incommensurable, but spatially periodic, with a period equal to 2π .

The STKD method is applied in the sampled square $0 \leq x \leq 10, 0 \leq t \leq 5$ (see Figure 12, right), where x and t are discretized using 50 and 25 points, respectively. The tunable parameters of the method are $\varepsilon_1 = 10^{-8}$, $\varepsilon_2 = 10^{-7}$, $d^x = 1$, and $d^t = 10$. The obtained $\omega - \kappa$ diagram is given in Figure 13, which shows that the pattern is a modulated TW, obtained as a linear superposition of two pure TWs, whose phase velocities are $c_1 = \omega_1/\kappa_1 = \pi/45$ and $c_2 = \omega_2/\kappa_2 = 1/\sqrt{10}$. The method provides the involved 12 wavenumbers and 4 frequencies with zero machine accuracy, while the

RRMS error (see (16)) of the reconstruction is $\sim 10^{-14}$. It is remarkable that these extremely good results are obtained using a sampled spatial interval that is somewhat larger to the spatial period, but a temporal sampled interval that is much smaller than the largest temporal period, which is $2\pi/\tilde{\omega}_1 = 45$. Also note that the extrapolation from the sampled time interval $0 \leq t \leq 5$ (Figure 12, right) to the larger time interval $0 \leq t \leq 100$ (Figure 12, left) still produces a good reconstruction, whose RRMS error, as defined in (16), is $\sim 10^{-14}$.

The sequential method has also been applied to analyze this toy model, obtaining results that are similar to those described above.

5. Applications of STKD

Let us now illustrate in some applications the ability of the STKD method to uncover the spatiotemporal structure of complex dynamics. Among these applications, we consider the CGLE (already analyzed with the temporal HODMD method in Section 3.1) and the dynamics of the thermal convection in a rotating spherical shell subject to gravitational force.

5.1. Application of the STKD Method to the CGLE. For the CGLE, the STKD counterpart of (27) is

$$A(x, t) = e^{i\omega_0 t} \sum_{mn} a_{mn} u_{mn} e^{i(\kappa_m x + \omega_n t)}, \quad (51)$$

where the modes u_{mn} are now unit complex numbers. Concentrating on the two cases analyzed in Section 3.1 and Figures 3 and 4, we use the same snapshots and tunable parameters used in Section 3.1, including the index d^t , which is set as being equal to the index d used in Section 3.1. The index d^x , instead, is set as $d^x = 1$. In both cases, the RRMS reconstruction errors, as defined in (16), are comparable to

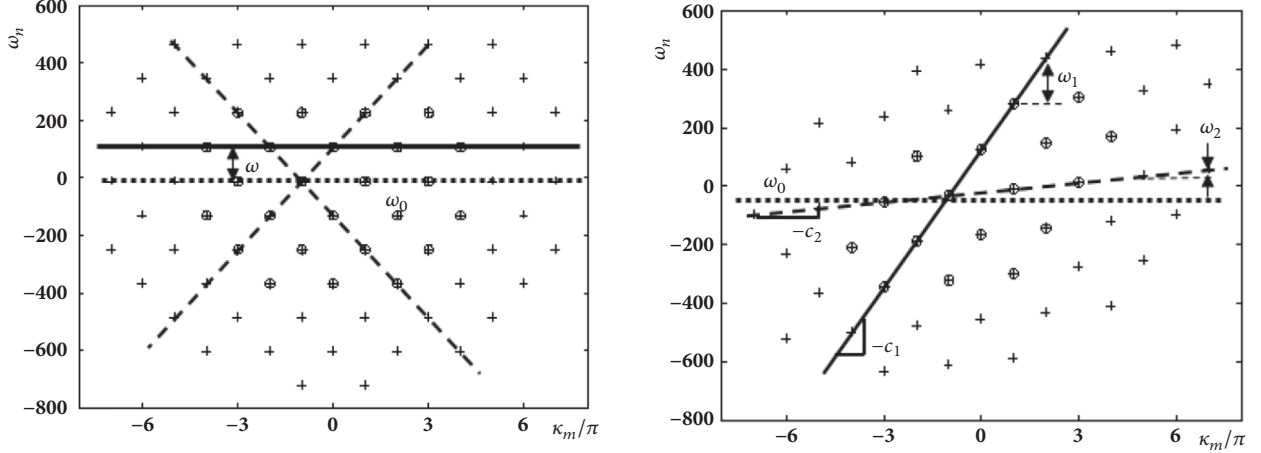


FIGURE 14: ω - κ diagrams resulting from the STKD method applied to the CGLE for the cases considered in Figure 3 (left plot) and Figure 4 (right plot). As in Figure 3, left, and Figure 4, left, crosses and circles label results obtained with high and low accuracies, respectively.

their counterparts obtained in Section 3.1 using the temporal HODMD method, retaining 25 and 71 spatiotemporal modes for low and high accuracy, respectively, in the case considered in Figure 3 and 17 and 49 modes for low and high accuracy, respectively, in the case considered in Figure 4. The ω - κ diagrams for these two cases are given in Figure 14. The left plot in this figure shows that points can be conformed in either horizontal or oblique straight lines, which clearly indicates that the pattern can be considered as either a modulated SW or a modulated TW, consistent with what is clearly seen in Figure 3. As anticipated in Section 3.1, the value of ω_0 in the expansion (51) equals 11.6, while the frequencies ω_n appearing in the expansion in the right hand side of (51) are positive and negative harmonics of the fundamental frequency $\omega_1 = 118.7$, which is associated with the wavenumber $\kappa_1 = \pi$. Thus, this pattern is slowly modulated (because ω_0 is small compared to ω_1) and when it is seen as a TW, the phase velocity is $c = \pm\omega_1/\kappa_1 = \pm 37.8$.

Concerning the right plot in Figure 14, the points in this plot are conformed in the indicated oblique straight lines, which means that the pattern is a modulated TW. As in Section 3.1, the frequency ω_0 appearing in (51) equals -21.2, while the various frequencies appearing in the expansion in the right hand side of this equations are all combinations of the fundamental frequencies $\omega_1 = 156.6$ and $\omega_2 = 22.2$, whose associated wavenumbers are $\kappa_1 = \pi$ and $\kappa_2 = 2\pi$, respectively; the associated phase velocities are $c_1 = -\omega_1/\kappa_1 = -49.8$ and $c_2 = -\omega_2/\kappa_2 = -3.53$, which are indicated in Figures 3 and 4, middle and right plots.

In summary, the STKD method computes quite well the frequencies and wavenumbers involved in SWs and TWs for the CGLE and permits straightforward identification of the associated phase velocities.

5.2. STKD in a Three-Dimensional Spherical Shell with Radial Thermal Convection, Subject to Radial Gravity. Let us consider the three-dimensional thermal convection in a rotating spherical shell, heated from the inside, subject to a radial

gravitational force. This problem is relevant in geophysical and astrophysical fluid dynamics [71], in connection with the transport of mass and energy in the upper atmospheres of stars and giant planets, where it is specially interesting the identification of periodic and quasi-periodic rotating waves promoted by convective instabilities.

The continuity, momentum, and energy equations governing the dynamics of the flow, using the Boussinesq approximation in a rotating (with angular velocity Ω) frame linked to the spherical shell, are

$$\nabla \cdot \mathbf{v}, \quad (52)$$

$$\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + 2E^{-1} \mathbf{k} \times \mathbf{v} = -\nabla \Pi + \nabla^2 \mathbf{v} + \Theta \mathbf{r}, \quad (53)$$

$$\sigma (\partial_t \Theta + \mathbf{v} \cdot \nabla \Theta) = \nabla^2 \Theta + R\eta (1 - \eta)^{-2} r^{-3} \mathbf{r} \cdot \mathbf{v}. \quad (54)$$

Here, \mathbf{v} , Π , and Θ are the velocity vector, the pressure, and the perturbation of the temperature from the quiescent purely convective state, respectively, $\mathbf{r} = (x, y, z)$ is the position vector referred to a Cartesian coordinate system, with the origin at the center of the sphere and the z axis along the axis of rotation, $r = |\mathbf{r}|$, and \mathbf{k} is the unit vector along the z axis. The boundary conditions are homogeneous Dirichlet for both the velocity and temperature at the inner and outer spheres.

The problem depends on three *nondimensional parameters*, the *Prandtl number* $\sigma = \nu/\kappa$ (where ν and κ are the kinematic viscosity and thermal conductivity, respectively), the *Eckman number* $E = \nu/(\Omega d^2)$ (where $d = r_o^* - r_i^*$ is the radial gap between the spheres), the *Rayleigh number* $R = \gamma \alpha \Delta T d^4 / (\nu \kappa)$ (where α is the thermal expansion coefficient, γr_i^* is the imposed radial gravity, and ΔT is increment in temperature), and the *inner to outer radii ratio* $\eta = r_i^*/r_o^*$. This problem is invariant under the symmetry group $SO(2) \times \mathbb{Z}_2$, generated by rotations about the z axis and up-down reflection on the equatorial plane. Invariance under rotation is essential to obtain TWs (which are rotating waves along the azimuthal direction in the present context), most of which are

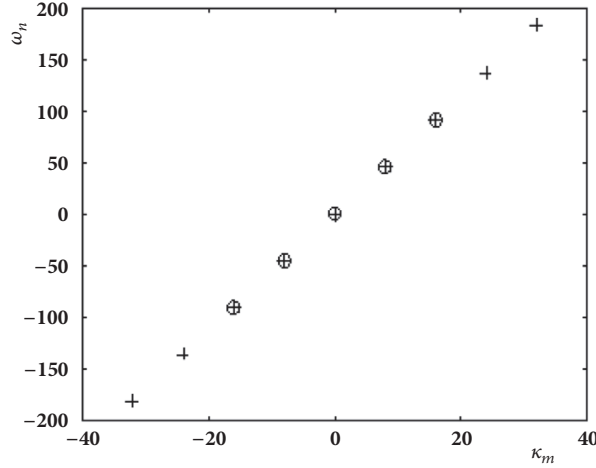


FIGURE 15: The ω - κ diagram (with circles and crosses indicating low and high accuracy) for the attractor at $R = 8 \cdot 10^5$.

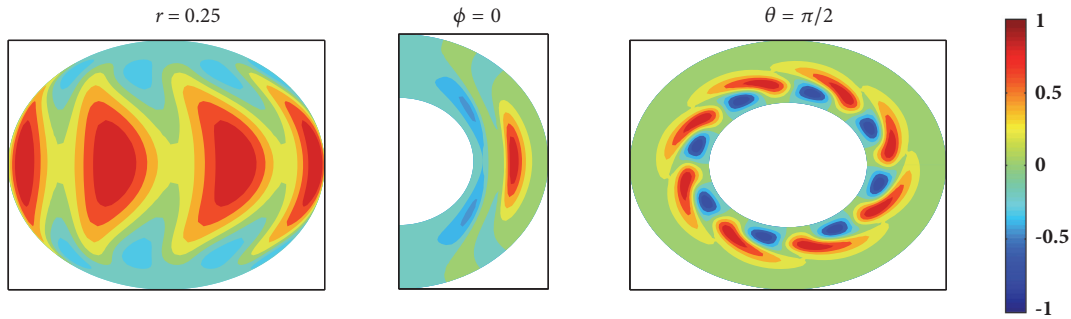


FIGURE 16: A snapshot showing the contour levels of Θ at the indicated surfaces for $R = 8 \cdot 10^5$.

also (instantaneously) invariant under the up/down reflection symmetry.

The computational domain is composed by 25, 128, and 32 grid points in the radial, colatitudinal, and azimuthal directions, respectively. The *numerical solver* used to solve this problem is described in [72–75], where the problem is reformulated by taking the curl of the momentum equation (which eliminates the pressure) and writing the solenoidal velocity field (which eliminates the continuity equation) as

$$\mathbf{v} = \nabla \times (\Psi \mathbf{r}) + \nabla \times \nabla \times (\Phi \mathbf{r}), \quad (55)$$

where Ψ and Φ are the *toroidal and poloidal scalar potentials*, respectively, first introduced by Chandrasekhar [76]. The numerical data used in the analysis below have been provided to us by Professors Marta Net and Joan Sánchez. These data correspond to $E = 10^{-4}$, $\eta = 0.35$, $\sigma = 0.1$, and two values of the Rayleigh number, namely, $R = 8 \cdot 10^5$ and $R = 1.1 \cdot 10^6$. In both cases, the STKD method is applied to the temperature field, which gives the expansion

$$\Theta(r_{j_1}, \theta_{j_2}, \phi, t) \approx \sum_{m,n=1}^{M,N} a_{mn} \Theta_{mnj_1j_2} e^{i(k_m \phi + \omega_n t)}, \quad (56)$$

where r_{j_1} and θ_{j_2} denote the mesh points in the radial and colatitudinal directions, respectively, and ϕ denotes the

azimuthal coordinate. As in the previous applications of the STKD method, two sets of values of the tunable parameters are considered, namely, $(\varepsilon_1, \varepsilon_2) = (10^{-4}, 5 \cdot 10^{-3})$ and $(10^{-5}, 5 \cdot 10^{-4})$, which will be referred to as low and high accuracy, respectively. The index d may be selected as $d = 1$ for both the spatial and temporal applications of the HODMD method that are needed in the second step of the STKD method.

For $R = 8 \cdot 10^5$, the STKD method is applied considering 100 equispaced snapshots in a time interval of length 0.3, selected after sufficiently large timespan to eliminate transient behavior. The RRMS reconstruction error, as defined in (16), is $1.6 \cdot 10^{-3}$ and $1.5 \cdot 10^{-5}$ for low and high accuracies, respectively, retaining 5 and 9 spatiotemporal modes, respectively. The $\omega - \kappa$ diagram is as given in Figure 15, which shows that for both low and high accuracy; the relevant points (whose frequencies and wavenumbers are multiples of $\omega_1 = 45.5$ and $\kappa_1 = 8$, respectively) are in a straight line passing through the origin. As explained in Section 4, this means that the pattern is a periodic, pure TW; namely, it is steady in a rotating frame with phase speed, $c = -\omega_1/\kappa_1 = -5.69$. Such a steady solution is considered in Figure 16, where the fundamental wavenumber $\kappa_1 = 8$ is clearly seen in the equatorial section $\phi = \pi/2$.

Concerning the case $R = 1.1 \cdot 10^6$, in which the dynamics are more complex than in the former case, the STKD method is applied considering 500 equispaced snapshots in a time

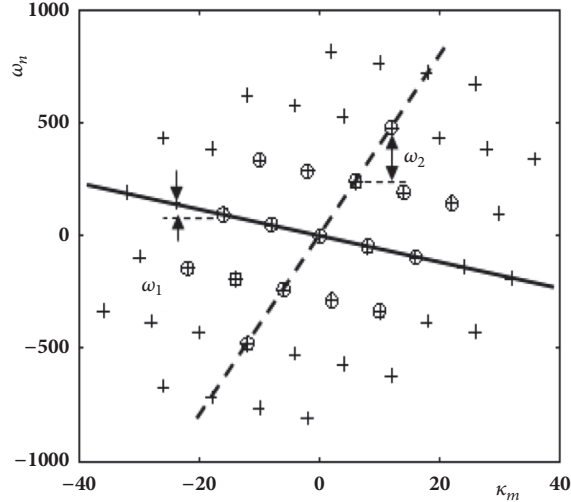


FIGURE 17: Counterpart of Figure 15 $R = 1.1 \cdot 10^6$.

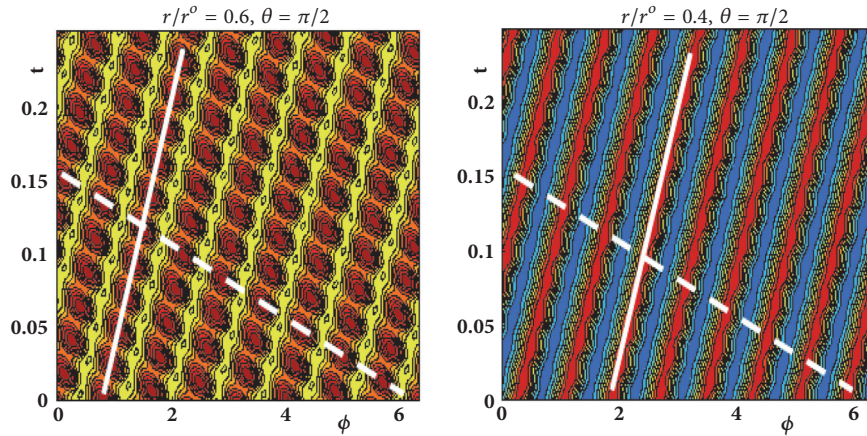


FIGURE 18: The quasi-periodic TW for $R = 1.1 \cdot 10^6$: color maps of Θ along the azimuthal coordinate at the indicated equatorial circles.

interval of length 1.4, again selected after sufficiently large timespan to eliminate transient behavior. Now, the RRMS reconstruction error, as defined in (16), is $8 \cdot 10^{-3}$ and $1.4 \cdot 10^{-4}$ for low and high accuracies, respectively, retaining 17 and 47 spatiotemporal modes, respectively. The ω - κ diagram is as given in Figure 17, where it can be seen that the relevant points are aligned along oblique straight lines. The fundamental frequencies and wavenumbers are $(\omega_1, \kappa_1) = (47.6, 8)$ and $(-239.8, 6)$, while the remaining points are linear combinations (with integer coefficients) of these. Thus, the pattern is a modulated TW, which is periodic in a frame moving with either of the phase velocities $c_1 = \omega_1/\kappa_1 = -5.95$ or $c_2 = \omega_2/\kappa_2 = 40$.

Since the counterpart of the space-time diagrams presented for the CGLE in Section 3.1 would lead to four-dimensional plots that are not possible in the present case, the character of the pattern as a modulated TW is illustrated in Figure 18 considering the restricted space-time diagrams (along the azimuthal coordinate) in two equatorial circles. As can be seen in these plots, the modulated TW character (with the already identified phase velocities, indicated here with straight lines) of the pattern is clearly seen.

5.3. Additional Applications. The STKD method has already been used in the analysis of TWs appearing in the following:

- (i) Off-shore wind turbines [64], whose structure is of great technological interest
- (ii) The three-dimensional cylinder wake [77], which permits identifying well the TWs associated with the already mentioned periodic modes A and B, and also with some quasi-periodic modes. These modes play a fundamental role in the associated dynamics [61]

Additional ongoing applications include the following:

- (i) Interacting wakes of bluff bodies, which generally gives complex quasi-periodic phenomena, including at least the frequencies and wavenumbers associated with the individual wakes
- (ii) Several turbomachinery configurations
- (iii) The reconstruction the 3D flow from various (PIV) 2D measurements, taking advantage of the fact that if the 3D flow is given by (34), then the whole unsteady 3D flow field may be reconstructed if the (steady) 2D

modes $\mathbf{u}_{mn}(\mathbf{y})$ are available. These can be computed by fitting with unsteady 2D measurements. Note that this is a demanding application of great scientific and technological interest

- (iv) The identification of the large scale dynamics in fully turbulent flows from either experimental (PIV) or numerical data resulting from direct numerical simulation. Again, this is a demanding application of great scientific interest

6. Conclusions

A review has been presented on the purely data-driven HODMD and STKD methods and their multiple applications. These methods decompose either temporal or spatiotemporal experimental or numerical signals as a combination of modes. In particular, HODMD is an extension of standard DMD that has general validity, namely it provides good results (identifying the relevant temporal growth rates and frequencies) in cases in which standard DMD fails. STKD also computes the relevant spatial growth rates and wavenumbers, and provides $\omega - \kappa$ diagrams that permit identifying TWs or SWs in the given data. These methods allow for two main tasks:

- (i) Uncovering the physics underlying the given data. In particular, in the case of experimental data, these methods are able to filter noise quite efficiently and extract patterns that were completely masked by experimental noise.
- (ii) Constructing ROMs whose online operation is very fast. This is because the HODMD and STKD reconstructions involve semianalytic descriptions that only require a limited amount of operations.

The HODMD and STKD methods have been illustrated in various applications of basic scientific interest, including the CGLE, the ZNMF jet, the thermal convection in a spherical rotating shell, and the wake of a circular cylinder. Several, more engineering oriented applications dealing with, e.g., aircraft flight tests and wind turbine problems, have also been considered.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness, under Grant TRA2016-75075-R.

References

- [1] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2nd edition, 1988.
- [2] J. Laskar, "Frequency analysis of a dynamical system," *Celestial Mechanics & Dynamical Astronomy*, vol. 56, no. 1-2, pp. 191-196, 1993.
- [3] J. Laskar, "Frequency analysis for multi-dimensional systems. Global dynamics and diffusion," *Physica D: Nonlinear Phenomena*, vol. 67, no. 1-3, pp. 257-281, 1993.
- [4] P. J. Schmid and J. L. Sesterhenn, "Dynamic mode decomposition of numerical and experimental data," in *Proceedings of the Bulletin of the American Physical Society 61st APS meeting, San Antonio*, p. 208, 2008.
- [5] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115-127, 2009.
- [6] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5-28, 2010.
- [7] J. H. Tu, C. W. Rowley, D. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391-421, 2014.
- [8] S. Le Clainche and J. M. Vega, "Higher order dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 882-925, 2017.
- [9] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130-141, 1963.
- [10] R. Temam, "Inertial manifolds," *The Mathematical Intelligencer*, vol. 12, no. 4, pp. 68-74, 1990.
- [11] S. Le Clainche and J. M. Vega, "Higher order dynamic mode decomposition to identify and extrapolate flow patterns," *Physics of Fluids*, vol. 29, Article ID 084102, 2017.
- [12] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, no. 3-4, pp. 579-616, 1991.
- [13] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and Turbulence*, D. A. Rand and L. S. Young, Eds., vol. 898 of *Lecture Note in Mathematics*, pp. 366-381, Springer, Berlin, Germany, 1981.
- [14] N. H. Packard, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Physical Review Letters*, vol. 45, pp. 712-716, 1980.
- [15] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. Nathan Kutz, "Chaos as an intermittently forced linear system," *Nature Communications*, vol. 8, no. 1, 2017.
- [16] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620-627, 1985.
- [17] D. S. Broomhead and R. Jones, "Time-series analysis," *Proceedings of the Royal Society A Mathematical, Physical and Engineering Sciences*, vol. 423, no. 1864, pp. 103-121, 1989.
- [18] D. Giannakis and A. J. Majda, "Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 7, pp. 2222-2227, 2012.
- [19] K. K. Chen, J. H. Tu, and C. W. Rowley, "Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses," *Journal of Nonlinear Science*, vol. 22, no. 6, pp. 887-915, 2012.
- [20] G. H. Golub and R. J. LeVeque, "Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems," vol. 3 of *ARO Rep. 79*, pp. 1-12, U. S. Army Res. Office, Research Triangle Park, N.C..
- [21] T. Askham and J. N. Kutz, "Variable projection methods for an optimized dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 380-416, 2018.

- [22] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, “Sparsity-promoting dynamic mode decomposition,” *Physics of Fluids*, vol. 26, no. 2, p. 024103, 2014.
- [23] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the Koopman operator: extending dynamic mode decomposition,” *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [24] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, “De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets,” *Theoretical and Computational Fluid Dynamics*, vol. 31, no. 4, pp. 349–368, 2017.
- [25] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, “Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition,” *Experiments in Fluids*, vol. 57, no. 3, 2016.
- [26] N. Takeishi, Y. Kawahara, Y. Tabei, and T. Yairi, “Bayesian dynamic mode decomposition,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 2814–2821, Australia, August 2017.
- [27] C. Dicle, H. Mansour, D. Tian, M. Benosman, and A. Vetro, “Robust low rank dynamic mode decomposition for compressed domain crowd and traffic flow analysis,” in *Proceedings of the 2016 IEEE International Conference on Multimedia and Expo, ICME 2016, USA, July 2016*.
- [28] S. Le Clainche and J. M. Vega, “Spatio-temporal Koopman decomposition,” *Journal of Nonlinear Science*, vol. 28, no. 5, pp. 1793–1842, 2018.
- [29] D. Giannakis, J. Slawinska, and Z. Zhao, “Spatiotemporal feature extraction with data-driven Koopman operators,” in *Proceedings of the in JMLR: Workshop and Conference*, vol. 44, pp. 103–115, 2015.
- [30] D. Giannakis, A. Ourmazd, J. Slawinska, and Z. Zhao, *Spatio-temporal pattern extraction by spectral analysis of vector-valued observables*, 2017, arXiv preprint 1711.02798.
- [31] A. S. Sharma, I. Mezić, and B. J. McKeon, “Correspondence between Koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier-Stokes equations,” *Physical Review Fluids*, vol. 1, no. 3, 2016.
- [32] B. Koopman, “Hamiltonian systems and transformations in Hilbert space,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, pp. 315–318, 1931.
- [33] I. Mezić, “Analysis of fluid flows via spectral properties of the Koopman operator,” *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.
- [34] M. C. Cross and P. C. Hohenberg, “Pattern formation outside of equilibrium,” *Reviews of Modern Physics*, vol. 65, pp. 851–1112, 1993.
- [35] M. Haragus and G. Iooss, *Local bifurcations, center manifolds, and normal forms in infinite-dimensional dynamical systems*, Universitext, Springer-Verlag London, Ltd., London; EDP Sciences, Les Ulis, 2011.
- [36] C. Gao, W. Zhang, J. Kou, Y. Liu, and Z. Ye, “Active control of transonic buffet flow,” *Journal of Fluid Mechanics*, vol. 824, pp. 312–351, 2017.
- [37] K. H. Park, S. O. Jun, S. M. Baek, M. H. Cho, K. J. Yee, and D. H. Lee, “Reduced-order model with an artificial neural network for aerostructural design optimization,” *Journal of Aircraft*, vol. 50, no. 4, pp. 1106–1116, 2013.
- [38] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [39] G. W. Stewart, “Error and perturbation bounds for subspaces associated with certain eigenvalue problems,” *SIAM Review*, vol. 15, pp. 727–764, 1973.
- [40] <https://github.com/LeClaincheVega/HODMD>.
- [41] S. Le Clainche, J. M. Vega, and J. Soria, “Higher order dynamic mode decomposition of noisy experimental data: The flow structure of a zero-net-mass-flux jet,” *Experimental Thermal and Fluid Science*, vol. 88, pp. 336–353, 2017.
- [42] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [43] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [44] I. S. Aranson and L. Kramer, “The world of the complex Ginzburg-Landau equation,” *Reviews of Modern Physics*, vol. 74, article 99, 2002.
- [45] T. Cebeci, *Convective heat transfer*, Horizons Publishing Inc., Long Beach, CA; Springer-Verlag, Berlin, 2002.
- [46] J. E. Carter and J. Soria, “The evolution of round zero-net-mass-flux jets,” *Journal of Fluid Mechanics*, vol. 472, pp. 167–200, 2002.
- [47] J. Soria, “Experimental studies of the near-field. Spatio-temporal evolution of zero-net-mass-flux (ZNMF) jets,” in *Vortex Rings and Jets, Fluid Mechanics and Its Applications*, vol. 111, pp. 61–92, Springer, 111, 2015.
- [48] A. Glezer and M. Amitay, “Synthetic jets,” in *Annual review of fluid mechanics, Vol. 34*, vol. 34 of *Annu. Rev. Fluid Mech.*, pp. 503–529, Annual Reviews, Palo Alto, CA, 2002.
- [49] H. Wang and S. Menon, “Fuel-air mixing enhancement by synthetic microjets,” *AIAA Journal*, vol. 39, no. 12, pp. 2308–2319, 2001.
- [50] B. L. Smith and A. Glezer, “Jet vectoring using synthetic jets,” *Journal of Fluid Mechanics*, vol. 458, pp. 1–24, 2002.
- [51] A. Pavlova and M. Amitay, “Electronic cooling using synthetic jet impingement,” *Journal of Heat Transfer*, vol. 128, no. 9, pp. 897–907, 2006.
- [52] L. N. Cattafesta III and M. Sheplak, “Actuators for active flow control,” *Annual Review of Fluid Mechanics*, vol. 43, pp. 247–272, 2011.
- [53] E. DeMont and J. Gosline, “Mechanics of jet propulsion in the hydromedusan jellyfish, *polyorchis penicillatus*,” *Journal of Experimental Biology*, vol. 134, pp. 347–361, 1988.
- [54] <https://nek5000.mcs.anl.gov/>.
- [55] C. H. K. Williamson, “Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers,” *Journal of Fluid Mechanics*, vol. 206, pp. 579–627, 1989.
- [56] C. H. K. Williamson, “Vortex dynamics in the cylinder wake,” *Annual Review of Fluid Mechanics*, vol. 28, pp. 477–539, 1996.
- [57] C. P. Jackson, “A finite-element study of the onset of vortex shedding in flow past variously shaped bodies,” *Journal of Fluid Mechanics*, vol. 182, pp. 23–45, 1987.
- [58] M. Provansal, C. Mathis, and L. Boyer, “Bénard-von Kármán instability: transient and forced regimes,” *Journal of Fluid Mechanics*, vol. 182, pp. 1–22, 1987.
- [59] D. Barkley and R. D. Henderson, “Three-dimensional Floquet stability analysis of the wake of a circular cylinder,” *Journal of Fluid Mechanics*, vol. 322, pp. 215–241, 1996.
- [60] D. Barkley, L. S. Tuckerman, and M. Golubitsky, “Bifurcation theory for three-dimensional flow in the wake of a circular cylinder,” *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 61, no. 5, part A, pp. 5247–5252, 2000.

- [61] H. M. Blackburn, F. Marques, and J. M. Lopez, "Symmetry breaking of two-dimensional time-periodic wakes," *Journal of Fluid Mechanics*, vol. 522, pp. 395–411, 2005.
- [62] J. M. Pérez, S. L. Clainche, and J. M. Vega, "Alternative three-dimensional instability analysis of the wake of a circular cylinder," in *Proceedings of the 8th AIAA Theoretical Fluid Mechanics Conference, 2017*, USA, June 2017.
- [63] G. Ramos, V. Beltrán, S. Le Clainche, E. Ferrer, and J. M. Vega, "Flow structures in the turbulent wake of a cross-flow wind turbine," in *Proceedings of the Wind Energy Symposium, 2018*, USA, January 2018.
- [64] S. Le Clainche, X. Mao, and J. M. Vega, "Spatio-temporal Koopman Decomposition in offshore wind turbines," in *Proceedings of the International Symposium on Transport Phenomena and Dynamics of Rotating Machinery (ISROMAC, Maui, Hawaii, 2017*.
- [65] V. Beltrán, S. Le Clainche, and J. M. Vega, "Characterization of the Wake past a Two-dimensional Multi-body Cylinder Arrangement," in *Proceedings of the 2018 AIAA Aerospace Sciences Meeting*, Kissimmee, Florida.
- [66] M. Harris, M. Hand, and A. Wright, "A Lidar for Turbine Control," Technical Report NREL/TP-500-39154, National Renewable Energy Laboratory (NREL), Golden, CO, USA, 2006.
- [67] C. J. Karlsson, F. Å. A. Olsson, D. Letalick, and M. Harris, "All-fiber multifunction continuous-wave coherent laser radar at 1.55 μm for range, speed, vibration, and wind measurements," *Applied Optics*, vol. 39, no. 21, pp. 3716–3726, 2000.
- [68] S. Le Clainche, L. Lorente, and J. M. Vega, "Wind Predictions Upstream Wind Turbines from a LiDAR Database," *Energies*, vol. 11, no. 3, pp. 543–558, 2018.
- [69] S. Le Clainche, F. Sastre, J. M. Vega, and A. Velázquez, "Higher order dynamic mode decomposition applied to post-process a limited amount of noisy PIV data," in *47th AIAA Fluid Dynamics Conference*, Denver, Co, 2017, AIAA paper 2017-3304.
- [70] S. Le Clainche, R. Moreno-Ramos, P. Taylor, and J. M. Vega, "New Robust Method to Study Flight Flutter Testing," *Journal of Aircraft*, pp. 1–8, 2018.
- [71] P. Charbonneau, "Stellar Dynamos," in *Solar and Stellar Dynamos*, vol. 39 of *Saas-Fee Advanced Courses*, pp. 187–214, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [72] F. Garcia, M. Net, B. García-Archilla, and J. Sánchez, "A comparison of high-order time integrators for thermal convection in rotating spherical shells," *Journal of Computational Physics*, vol. 229, no. 20, pp. 7997–8010, 2010.
- [73] J. Sanchez, M. Net, and C. Simo, "Computation of invariant tori by Newton-Krylov methods in large-scale dissipative systems," *Physica D: Nonlinear Phenomena*, vol. 239, no. 3-4, pp. 123–133, 2010.
- [74] J. Sanchez and M. Net, "A parallel algorithm for the computation of invariant tori in large-scale dissipative systems," *Physica D: Nonlinear Phenomena*, vol. 252, pp. 22–33, 2013.
- [75] J. Sánchez, F. Garcia, and M. Net, "Computation of azimuthal waves and their stability in thermal convection in rotating spherical shells with application to the study of a double-Hopf bifurcation," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 87, no. 3, 2013.
- [76] S. Chandrasekhar, *Hydrodynamic and Hydromagnetic Stability*, Oxford University Press, London, UK, 1961.
- [77] S. Le Clainche, J. M. Pérez, and J. M. Vega, "Spatio-temporal flow structures in the three-dimensional wake of a circular cylinder," *Fluid Dynamics Research. An International Journal*, vol. 50, no. 5, 051406, 19 pages, 2018.

Research Article

Computer Vision with Error Estimation for Reduced Order Modeling of Macroscopic Mechanical Tests

Franck Nguyen,¹ Selim M. Barhli,² Daniel Pino Muñoz,³ and David Ryckelynck ¹

¹Centre des Matériaux, Mines ParisTech PSL Research University, Evry 91003, France

²Safran Analytics, rue des Jeunes Bois, Châteaufort, CS 80112, 78772 Magny les Hameaux Cedex, France

³CEMEF, Mines ParisTech PSL Research University, CS 10207, 06904 Sophia Antipolis Cedex, France

Correspondence should be addressed to David Ryckelynck; david.ryckelynck@mines-paristech.fr

Received 29 June 2018; Accepted 13 November 2018; Published 2 December 2018

Academic Editor: Francisco J. Montáns

Copyright © 2018 Franck Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, computer vision enables recommending a reduced order model for fast stress prediction according to various possible loading environments. This approach is applied on a macroscopic part by using a digital image of a mechanical test. We propose a hybrid approach that simultaneously exploits a data-driven model and a physics-based model, in mechanics of materials. During a machine learning stage, a classification of possible reduced order models is obtained through a clustering of loading environments by using simulation data. The recognition of the suitable reduced order model is performed via a convolutional neural network (CNN) applied to a digital image of the mechanical test. The CNN recommend a convenient mechanical model available in a dictionary of reduced order models. The output of the convolutional neural network being a model, an error estimator, is proposed to assess the accuracy of this output. This article details simple algorithmic choices that allowed a realistic mechanical modeling via computer vision.

1. Introduction

In biomechanics, computer vision and mechanical testing have been coupled to obtain patient-specific simulation approaches, as proposed in [1]. At the same time, with the growth of industry 4.0, imaging techniques are more and more widespread in factories. When combined with artificial neural networks, digital images enable the classification of products to obtain the best possible process, as proposed in [2] for olive batches classification in oil extracting process or as shown in [3] for composite materials manufacturing. Nowadays, we have the possibility of extending these methods to the classification of mechanical parts produced in industry, in order to develop part-specific decision approaches. For mechanical parts, the quality of manufacturing processes has a direct influence on the ultimate mechanical properties of the manufactured parts. For example, the way the fracture is initiated in a specimen often reveals defects in the material whose origin can be tracked back to the manufacturing process [4]. In general, the numerical computation of mechanical stresses in a given

manufactured part allows the predictive evaluation of the link between the ultimate mechanical properties of this part and the manufacturing process. The reader can find an example of how to optimize a process for curing composite parts in [5] according to this paradigm. The mechanical modeling of manufactured parts has for purpose to verify if defects induced by a manufacturing process are tolerable, if an observed part must be rejected, or if the manufacturing process must be improved.

In this paper, we restrict our attention to the stress prediction in a part under an observed loading environment by a digital image, while including all its geometrical defects. We propose a hybrid approach that simultaneously exploits a data-driven model and a physics-based model, in mechanics of materials. The reader can find a review on hybrid modeling in [6] for remaining useful life predictions of engineering systems. We show that the strength of the proposed hybrid modeling is its ability to incorporate an error estimator related to the modeling chain with computer vision and convolutional neural networks (CNN) [7, 8].

As explained in [9], computer vision with deep convolutional neural networks has achieved state-of-the-art performance on standard recognition datasets and tasks. In this paper, we explore the capabilities of a CNN as a recommender system for the mechanical modeling of structures submitted to various loading. The proposed hybrid modeling couples a noncentered principal component analysis (PCA) and a CNN, in order to preserve an accurate description of spatial information.

Image processing for computer vision is usually very fast. It does not make sense to couple computer vision with numerical simulations of mechanical stresses that take hours of computation. Hence, we couple computer vision with reduced order modeling of structures, in order to get fast mechanical predictions of stresses.

A reduced order model is a surrogate model obtained by the projection of high dimensional equations on a reduced space, and it also involves a reduced approximation space for the variables of the high dimensional problem. When they are the same, the surrogate model is a Galerkin reduced order model [10]. In hyperreduced order models these two reduced spaces are different [11]. Then, because we consider projection of physics-based equations, hyperreduced order models preserve the physical parameters involved in the high dimensional equations. The reduced spaces involved in the hyperreduced modeling are spanned by empirical modes extracted from simulation data, by using the proper orthogonal decomposition [10] of known finite element predictions. This procedure is similar to a noncentered principal component analysis (PCA). Hence, the proposed modeling via computer vision exploits both simulation data and observational data, which are, respectively, finite element predictions and digital images of mechanical tests.

In general, the empirical modes obtained by noncentered PCA are very sensitive to the loading environment imposed when computing the simulation data. If the variety of loading conditions considered to calculate simulation data is too wide, the number of empirical modes becomes too large. They can no longer reduce the numerical complexity of the mechanical balance equations. Clustering methods have been applied for model-order reduction in [12, 13], in order to preserve small reduced-bases of empirical modes. Moreover, cluster-based reduced order modeling (CROM) has been proposed in [14] to define a small subset of critical data to learn an efficient (CROM) with a sparse approximation space [15]. In this paper, a dictionary of hyperreduced order models is generated by considering clusters of possible loading environments in the observed mechanical tests. Then, the identification of the hyperreduced order model is done via recognition of a class of mechanical loads by a convolutional neural network. In practice, each item of the dictionary is not directly a hyperreduced order model. In order to face a possible variability on the geometry of the observed structures, it is more robust to define an item of the dictionary as a set of finite element solutions for various ideal geometries and for a given class of mechanical loads. Hence, the proposed workflow is robust enough to face geometrical defects in the observed mechanical parts. We assume that the mesh for the finite element modeling of the parts is obtained by

using image-meshing techniques, as proposed in [16], of segmented 3D digital images obtained by X-ray computed tomography. We refer the reader to [17, 18] for more details on finite element modeling of 3D images obtained by X-ray computed tomography. An example of X-ray computed tomography applied to manufactured parts can be found in [19]. The ideal geometries involved in the proposed workflow were obtained by using computer-aided-design (CAD). Such CAD models are usually used to find an optimal design of the manufactured parts, with parametric finite element simulations [20].

2. Materials and Methods

Prior to the stress prediction, a reduced order model is setup for the projection of the mechanical balance equations. Here the reduced order model concerns the displacement in the observed mechanical part. Usually, for the computation of reduced approximations in nonlinear problems, we formally consider all possible situations in a given parameter space [21]. The parameters aim to describe all the possible mechanical problems, in advance, before the observation of a realization of one situation. This approach defines a tensor for the description of all possible displacements. The order of this tensor is the number of scalar parameters involved in the parametric equations, plus one. For instance, if a single parameter is introduced then we need two indices i and j to have access to the value of the i^{th} degree of freedom of a finite element model for the j^{th} value of the parameter. When D parameters are introduced, we need $D+1$ indices: $i, j^{(1)}, \dots, j^{(D)}$, to have access to a scalar value in the tensor containing all the possible displacements. This tensor formalism aims to introduce a sampling procedure of the parameter space in order to get an estimation of the reduced approximation, or a reduced basis, for the displacements. For instance, this sampling procedure can be achieved by the proper generalized decomposition (PGD) [21] or the Tensor Train decomposition [22, 23]. In this paper, the loading environment is depicted by an image of 3968×2976 pixels. Then the parameter space dimension is around 12 millions (the number of pixels) for the description of all possible loading environments. Hence the tensor formalism for model-order reduction would require the decomposition of a tensor of order 12 millions. To our knowledge, no tensor decomposition method has been applied to such a huge tensor order. A purely tensor approach seems to be unaffordable. In this paper, we do not pretend to model all the possible solutions of mechanical equations related to a huge parameter space. We do not follow the usual paradigm of low rank approximations. The proposed image-based modeling aims to exploit available data for fast approximate predictions with fast error estimation.

The workflow of the proposed modeling via computer vision is shown in Figure 1. Four kinds of inputs are required:

- (i) a 2D digital image of the part in the test machine, this image is denoted by I^* ;
- (ii) a database where are saved all simulation data, ordered with respect to a cluster index β and the index

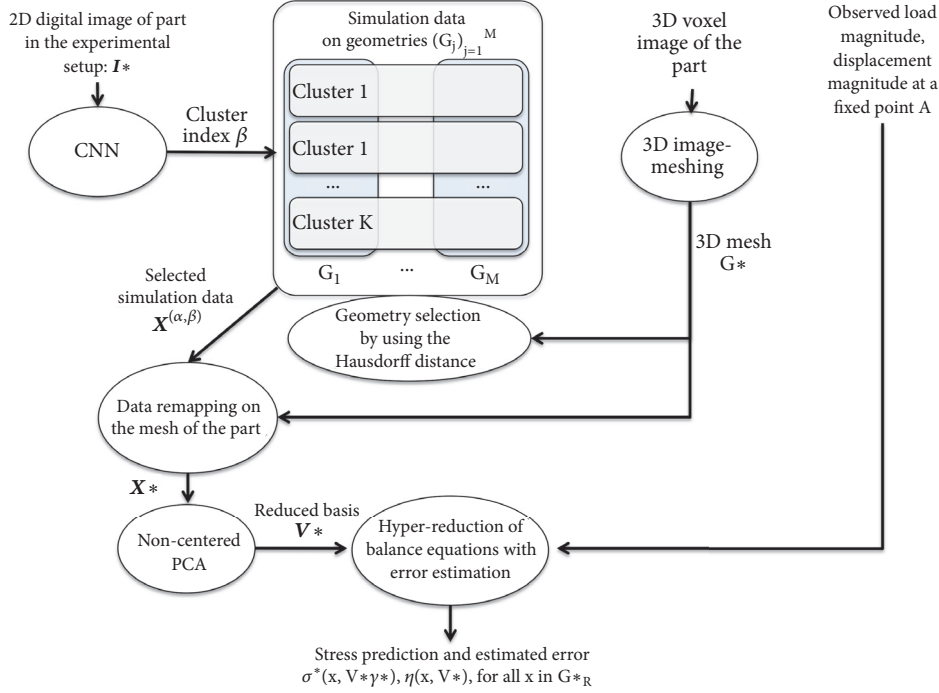


FIGURE 1: Workflow of the reduced order modeling via computer vision. The inputs are on the top of the figure; the outputs are the stress prediction $\sigma(x, V^* \gamma^*)$ by using an hyperreduced order model and an error estimator η related to the accuracy of the reduced basis V^* .

of the available ideal mesh in the list $(G_j)_{j=1}^M$, these are the meshes used when generating the simulation data by the finite element method;

- (iii) a 3D voxel image of the part alone, we assumed that this 3D image is obtained by X-ray computed tomography;
- (iv) a measurement of the load magnitude at the end of the mechanical test and a measurement of the displacement at one fixed point of on the part, at the end of the test.

The CNN network aims to recognize the index β of the class of loading environment. It gives access to the simulation data required to create on the fly a hyperreduced order model. The measurement of the displacement magnitude helps to get a precise location of the load by solving an inverse problem with hyperreduced equations as proposed in [24].

The stress predicted by the hyperreduction is obtained via constitutive equations in the framework of elasticity. It depends on the spatial position in the part, denoted by x , and nodal values of the displacement, denoted by the vector \mathbf{q}^* . This vector has d^* components. d^* is the order of the finite element model of the part. The superscript $*$ is introduced for the variables related to the observed mechanical part in the experimental setup. The mesh of the mechanical part is denoted by G^* and the stress is denoted by $\sigma(x, \mathbf{q}^*)$. The reduced approximation of the displacement reads $\mathbf{q}^* \approx V^* \boldsymbol{\gamma}^*$, where V^* is the matrix form of the reduced basis (it has less columns than rows) and $\boldsymbol{\gamma}^*$ is the solution

of the hyperreduced balance equations. Since the finite element method is a numerical scheme for partial differential equations, Dirichlet boundary conditions are applied on the boundary of the domain occupied by the mesh. Here these conditions are null. Hence the displacements belong to a vector space, which is a subspace of a Hilbert space. The columns of V^* span a subspace of this vector space. They fulfill the Dirichlet boundary conditions applied on G^* . The hyperreduced balance equations are set on a reduced mesh G_R^* , which is the restriction of G^* to the finite elements connected to a given list of degrees of freedom, denoted by F . The residual of the finite element equations is denoted by $\mathbf{r}^*(\mathbf{q}^*)$. Hence the hyperreduced balance equation reads [20]

$$\mathbf{V}^{*T} [F, :] \mathbf{r}^*(\mathbf{V}^* \boldsymbol{\gamma}^*) [F] = 0 \quad (1)$$

The larger the set F the higher the computational complexity of the projection of the equations when considering the observed geometry. In the sequel, F is the set of degrees of freedom (dof) indices near the loading areas on G^* supplemented by the list of dof in a region of interest. When V^* is the identity matrix and F contains all the dof of the mesh, then (1) returns to the original finite element equations, $\mathbf{r}^*(\mathbf{q}^*) = 0$.

Property 1. The following property holds: if the finite element solution is unique, if the solution of the hyperreduced equation is unique, and if the reduced basis is exactly reproducing the finite element solution, the hyperreduced solution is exact. Hence the following expression holds: $\mathbf{q}^* = V^* \boldsymbol{\gamma}^*$.

In hybrid hyperreduced order models proposed in [24, 25], the reduced basis is extended with few finite element shape functions available inside G_{R^*} . By construction, these shape functions are not connected to the remaining elements of G^* . The set of indices of these shape functions is denoted by P . It is a subset of F , by construction. For the sake of simplicity, we order the degrees of freedom such that $P = \{1, \dots, \text{card}(P)\} \subset F$, where $\text{card}(P)$ is the number of elements in the set P . Hence the reduced basis of the hybrid hyperreduced order model, denoted by \mathbf{V}^H , is the following block matrix:

$$\mathbf{V}^H = \begin{bmatrix} I^P & \\ & \mathbf{V}^* \\ 0 & \end{bmatrix} \quad (2)$$

where I^P is the identity matrix in dimension $\text{card}(P)$. By substituting \mathbf{V}^H for \mathbf{V}^* in the hyperreduced balance equations, one obtains the hybrid hyperreduced equations, of which solution is denoted by γ^H . If the projection of \mathbf{q}^* on \mathbf{V}^* is exact, then it has also an exact projection on the larger subspace spanned by \mathbf{V}^H . According to Property 1, if \mathbf{V}^* is exact and if the hybrid hyperreduced equations have a unique solution, then the first components of γ^H should be zero and the last one should be equal to γ^* . It turns out that if \mathbf{V}^* is exact and if the hybrid hyperreduced equations have a unique solution, then the stress predictions $\sigma(x, \mathbf{V}^* \gamma^*)$ and $\sigma(x, \mathbf{V}^H \gamma^H)$ are equal. Hence, the following error estimator is proposed, in order to assess the accuracy of the hyperreduced order model that has been recognized by the computer vision workflow:

$$\eta(x, \mathbf{V}^*) = c \left\| \sigma(x, \mathbf{V}^* \gamma^*) - \sigma(x, \mathbf{V}^H \gamma^H) \right\|, \quad x \in \Omega_R \quad (3)$$

where Ω_R is the spatial domain occupied by the reduced mesh G_{R^*} and c is a constant. The larger G_{R^*} , the more complex the error estimation. When $\sigma(x, \mathbf{V}^H \gamma^H)$ fulfills the finite element equilibrium equations, this error indicator is similar to the error indicator proposed in [26] for standard materials. If $c = 1$, if Ω_R is the domain occupied by the full mesh G^* , and if P contains all the dof indices, then $\eta(x, \mathbf{V}^*)$ is the true error. The constant c can be evaluated by following the procedure proposed in [25]. Here, we assume that $c = 1$.

As shown in Figure 1, \mathbf{V}^* is obtained by a noncentred PCA applied on simulation data \mathbf{X}^* . These data are displacement fields $\mathbf{X}^{(\alpha, \beta)}$ remapped on mesh G^* from the mesh G_α and restrained to the loads in the class β of the load clustering. This clustering of loading environments in the mechanical tests is presented below. A simple interpolation of the data in $\mathbf{X}^{(\alpha, \beta)}$ is done for the remapping of the displacement fields on the mesh G^* for nodes in the domain occupied by G_α . For nodes in G^* that are not in the domain occupied by the mesh G_α , the remapping is done via Laplace's equation with an enforced continuity at the boundary of G^* and outside G_α . In practice, a robust model reduction is achieved if the meshes $(G_j)_{j=1}^M$ are restricted to elements

that are not connected to nodes submitted to a concentrated load or a Dirichlet boundary condition. In the remapping procedure, the Dirichlet boundary conditions are enforced as Dirichlet boundary conditions of the Laplace's equation. Then $\mathbf{X}^{(\alpha, \beta)}$ fulfills the Dirichlet boundary condition on the mesh G^* .

The clustering of loading environments in the mechanical tests is performed by using the simulation data mentioned in the workflow in Figure 1. Unfortunately, these data do not have the same dimension since they are supported by different meshes $(G_j)_{j=1}^M$. Then, they are remapped on the mesh of a bounding box that surrounds all the meshes $(G_j)_{j=1}^M$, so comparisons are easier. The extrapolation of the data outside the meshes $(G_j)_{j=1}^M$ follows the Laplace's equation again. The remapped simulation data on the bounding box are saved in the tensor $\overline{\mathbf{X}} \in \mathbb{R}^{L \times \overline{d} \times M \times \overline{M}}$ of order 4. Four indices are introduced to have access to scalar values saved in $\overline{\mathbf{X}}$. This value is denoted by $\overline{\mathbf{X}}(i, p, j, n)$, where i is the load case index, p is the dof index in the bounding box, j is the index of the mesh in the list $(G_j)_{j=1}^M$, and n is the index related to additional parameter variations. For instance, local variations of the mechanical properties have the capability to enrich the simulation data for model-order reduction as proposed in [11]. When considering simulation data related to finite element models, a high resolution in spatial fields is achieved by high dimensional finite element space. Hence \overline{d} , the second dimension of $\overline{\mathbf{X}}$, is often larger than 10^5 and can reasonably be up to 10^7 in industrial applications. In very high-dimension spaces, all the data are "far away" from the centre. Hence, a feature extraction is required in the framework of mechanical modeling, prior to clustering the simulation data. Several tensor decomposition methods are available in the literature for feature extraction. For instance, the k-PCA has been coupled to the proper generalized decomposition method in [27, 28] for extracting hidden model parameters. We refer the reader to [29] for a review on feature extraction. Here, we adopt a hierarchical Tucker format [30]:

$$\overline{\mathbf{X}}(i, p, j, n) = \sum_{s=1}^{\widehat{d}} \widehat{\mathbf{V}}_{ps} \widehat{\mathbf{X}}(i, s, j, n), \quad \widehat{d} \leq \overline{d}, \quad (4)$$

where $\widehat{\mathbf{V}}$ is obtained by the following truncated singular value decomposition:

$$\begin{aligned} \widehat{\mathbf{A}} &= \widehat{\mathbf{V}} \widehat{\mathbf{S}} \widehat{\mathbf{W}}^T, \\ \widehat{\mathbf{S}}_{ii} &> \widehat{\mathbf{S}}_{i-1, i-1} > 0, \end{aligned} \quad (5)$$

where $\widehat{\mathbf{A}}$ is the reshape of $\overline{\mathbf{X}}$ as a second order tensor:

$$\begin{aligned} A_{pk} &= \overline{\mathbf{X}}(i, p, j, n), \\ k &= (i-1)M\overline{M} + (j-1)\overline{M} + n \end{aligned} \quad (6)$$

We restrict the clustering of data to features in dimension 2, for the ease of results visualization, and on the two following average data:

$$\begin{aligned}\overline{\widehat{X}}_i^{(1)} &= \frac{1}{M\overline{M}} \sum_{n=1}^{\overline{M}} \sum_{j=1}^M \widehat{X}(i, 1, j, n), \\ \overline{\widehat{X}}_i^{(2)} &= \frac{1}{M\overline{M}} \sum_{n=1}^{\overline{M}} \sum_{j=1}^M \widehat{X}(i, 2, j, n)\end{aligned}\quad (7)$$

This two-dimensional feature space enables visualizing clusters on loading environments. Here, we arbitrary select K clusters by using the k-means method. In the future we would experiment other clustering methods such as graph-based clustering, for instance. Then, we obtain a partition of the simulation data into K sets C_1, \dots, C_K . Each set is a list of loading indices in $\{1, \dots, L\}$. Then the original simulation data related to the cluster C_β and the mesh G_α are $\overline{X}(i, p, \alpha, n)$, for i in C_β , p in $\{1, \dots, \overline{d}\}$ and n in $\{1, \dots, \overline{M}\}$.

The CNN architecture chosen for this work is based on the layer composition initially described in [31]. The input to the CNN is a fixed-size RGB image of 246×246 pixels, denoted h^0 . This input image is obtained by downscaling the original 3968×2976 images denoted by I^* in Figure 1.

The image goes through a set of convolutional layers followed by max-pooling layers. The convolution and pooling stack is repeated 3 times. The rectified linear unit (ReLU) activation function [8] is used for the convolutional layers. This function is denoted by $\langle \bullet \rangle_+$. Each layer generates a feature map, denoted by h^k after the k^{th} layer. Each feature map is a tensor of order 3, which dimensions are denoted by N_1^k, N_2^k, N_3^k . The first feature map is h^0 . The k^{th} convolution layer applies linear filters to h^{k-1} . Each linear filter is determined by the weights W^k and the bias b_τ^k , such that

$$\begin{aligned}h^k(i, j, p) &= \left\langle \sum_{\alpha=1}^3 \sum_{\beta=1}^3 W^k(\alpha, \beta, \tau) \right. \\ &\quad \left. \cdot h^{k-1}(i+2-\alpha, j+2-\beta, l) + b_\tau^k \right\rangle_+\end{aligned}\quad (8)$$

where $p = (\tau - 1)N_3^{k-1} + l$, $\tau = 1, \dots, m_F^k$, m_F^k is the number of filters in the k^{th} layer, $i = 2, \dots, N_1^{k-1} - 1$, $j = 2, \dots, N_2^{k-1} - 1$, and $l = 1, \dots, N_3^{k-1}$. The first convolution layer uses 32 filters; the next two use 64 filters. Here, the convolution kernel size is set to 3×3 and the stride to 1 pixel, no padding is used. The max-pooling is performed with windows of 2×2 pixels and a stride of 2. It reads

$$h^k(i, j, p) = \max_{\alpha, \beta \in \{0, 1\}} h^{k-1}(i' + \alpha, j' + \beta, p) \quad (9)$$

where $i' = 2i - 1$, $j' = 2j - 1$. The output of the last pooling layer is flattened and fed through 3 fully connected layers: the

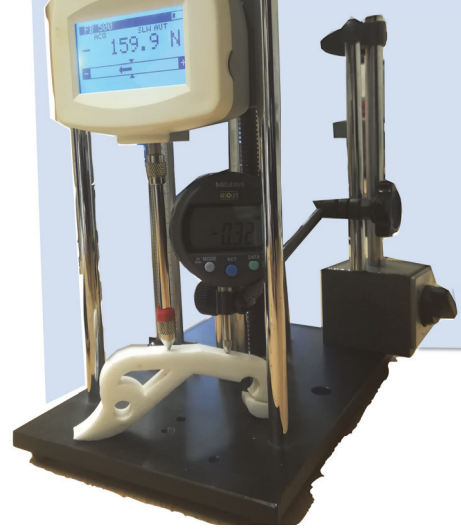


FIGURE 2: Experimental setup. A mechanical load of 159.9 N is applied on the top of the part in white. A displacement is measured at a fixed point.

first one has 512 nodes and the second one 64 nodes; those first two layers use the ReLu activation function. The third layer is a soft-max layer with 4 nodes thus performing the 4-way classification task, such that

$$h_i^k = \frac{e^{h_i^{k-1}}}{\sum_{j=1}^4 e^{h_j^{k-1}}}, \quad i = 1, \dots, 4 \quad (10)$$

In order to reduce overfitting, dropout regularization is implemented for the first fully connected layer with a dropout rate of 50%.

3. Results and Discussion

3.1. Experimental Setup. We consider a very simple mechanical test on a part in order to check its manufacturing process via the stress distribution in the part and the related response of the part submitted to various loading environments on the top of it. An image of the experimental setup is shown on Figure 2. The modeling via computer vision aims to recognize the loading environment applied on the part, in order to predict stresses by using a hyperreduced order model. The magnitude of the load and the vertical displacement at a fixed point are measured precisely during the mechanical test. But the location of the load has to be determined by the computer vision approach. We are considering $L=18$ possible loading cases regularly spaced on the top of the part.

The region of interest shown in red in Figure 3 is hidden in Figure 2 by the experimental setup. If this region of interest would be visible on each image of the experimental design, a digital image correlation [32] approach could have provided an estimate of the mechanical stresses in the part. But it is not appropriate here, because the part is partially masked in some digital images.

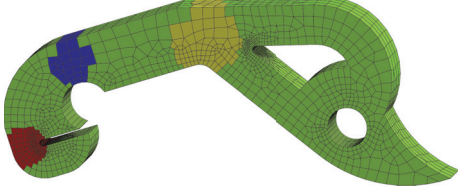


FIGURE 3: The red, blue, and yellow regions have been submitted to variations of the Young modulus in order to enhance the space spanned by the simulation data. The region in red is the region of interest selected by the designer of the experimental setup. The mesh shown here is the mesh G_4 . It has $5 \cdot 10^5$ degrees of freedom.

3.2. *Simulation Data, Feature Extraction, and Clusters of Mechanical Loading.* Prior to starting the manufacturing process, the experimental setup has been designed by using finite element simulations on four ideal geometries ($M=4$). The material of the part is elastic, for each mesh G_j . The mechanical constitutive equations are the following:

$$\frac{1+\nu}{E} \boldsymbol{\sigma}(\mathbf{x}, \mathbf{q}^j) - \frac{\nu}{E} \text{tr}[\boldsymbol{\sigma}(\mathbf{x}, \mathbf{q}^j)] \mathbf{I} = \sum_{i=1}^d \boldsymbol{\varepsilon}_i(\mathbf{x}) q_i^j \quad (11)$$

where $\boldsymbol{\varepsilon}_i$ is the symmetric part of the i^{th} finite element shape function related to the i^{th} degree of freedom of the mesh, E is Young's modulus, and ν is Poisson's coefficient. The mechanical properties of the material are $E = 1,600$ MPa, $\nu = 0.3$. Here, the residual of the finite element equilibrium equation reads

$$r_i^j(\mathbf{q}^j) = \int_{\Omega(G_j)} \boldsymbol{\varepsilon}_i(\mathbf{x}) : \boldsymbol{\sigma}(\mathbf{x}, \mathbf{q}^j) dx + F \delta_{i\theta} \quad (12)$$

Here, θ is the dof index where the load F is applied downward, $\Omega(G_j)$ is the spatial domain occupied by the mesh G_j , and $\delta_{i\theta}$ is the Kronecker delta. The observed magnitude of the mechanical load is 159.9 N.

For each geometry, three local variations of the Young modulus of -20% have been simulated by the finite element model, in order to enhance the space spanned by using the simulation data. The regions affected by these variations are shown in Figure 3. It turns out that we have done $\widetilde{M}=4$ predictions of the displacement fields, for each geometry and each loading case. The region of interest is shown in red in Figure 3. It covers the subdomain where stress concentrations are expected to be highest during the mechanical test.

An example of the remapping of a vertical displacement field onto the bounding box is shown on Figure 4. The mesh of the bounding box has $\widetilde{d}=8 \cdot 10^5$ dofs. The first feature mode (the first column of $\widehat{\mathbf{V}}$) is shown in the right of Figure 4. This feature mode is consistent with the bending simulation of the mechanical parts.

We have arbitrary chosen $K=4$ centroids to cluster the mechanical loadings. The L points ($\widehat{\mathbf{X}}_i^{(1)}, \widehat{\mathbf{X}}_i^{(2)}$) are shown in Figure 5 with the 4 clusters presented by red circles.

3.3. *Observational Data.* About 250 high resolution digital images for each class of loading have been generated before starting any mechanical experiment. Examples of such images are shown on Figure 6. It takes approximately 10 minutes to get 1000 images. The size of each image is 3968×2976 pixels.

3.4. *Training and Testing the Convolutional Neural Network.* The CNN has been implemented with Keras library in TensorFlow. The train/test set was built following a 90/10 ratio upon the 1000 digital images of loading environments. The volume of training data was artificially increased by using a synthetic data augmentation strategy. Each "original" image was transformed using a combination of random shear, zoom, and horizontal flip values. The shear was limited to a maximum of 10% and the zoom to a maximum of 30%.

The training was performed by optimizing a multinomial cross entropy loss using a minibatch gradient descend approach with the RMSprop adaptive learning rate method; batch size is set to 32 and the model is trained on 120K steps (60 epochs).

The performance of the CNN is assessed on the test set; a top-1 error value of 1.9% was achieved. This great performance is explained by the easiness of the classification task due to no large variations between input images being observed, since they are all related to the same experimental setup.

3.5. *Hyperreduction of Finite Element Equations.* The modeling via computer vision has been applied to a mechanical test. The mesh G^* of the part is in red in Figure 7. The smallest Hausdorff distance between the ideal meshes and G^* is achieved by G_4 , so $\alpha=4$. This is the grey mesh superposed to G^* in Figure 7. There is $d^*=8 \cdot 10^5$ dof in G^* .

The digital image of the mechanical part in the experimental setup is shown in Figure 8. The convolutional neural network recognizes the class number 3 of loading environment. Then $\beta=3$, and we can extract the simulation data from the database in order to compute the reduced basis \mathbf{V}^* , by using the noncentred PCA. We restrict this reduced basis to five empirical modes. Hence, the projection error of the selected simulation data on this reduced basis is less than 0.1%.

The reduced mesh G_{R^*} is generated on the fly. It is shown in Figure 9. Here, F is a set of dof in the region of possible loading for class #3, plus the dof close to the displacement measurement, plus the dof in the region of interest. The elements of G_{R^*} are the elements connected to F .

3.6. *Stress Prediction and Error Estimation.* A finite element simulation takes 45 min. The stress prediction by the hyperreduced order model shown in Figure 10 is obtained in less than 10 min. 99% of this computational time is spent for the solution of (1). By choosing a smaller set of degrees of freedom F , one can obtain prediction in less than 2 min. The larger the set F , the longer the

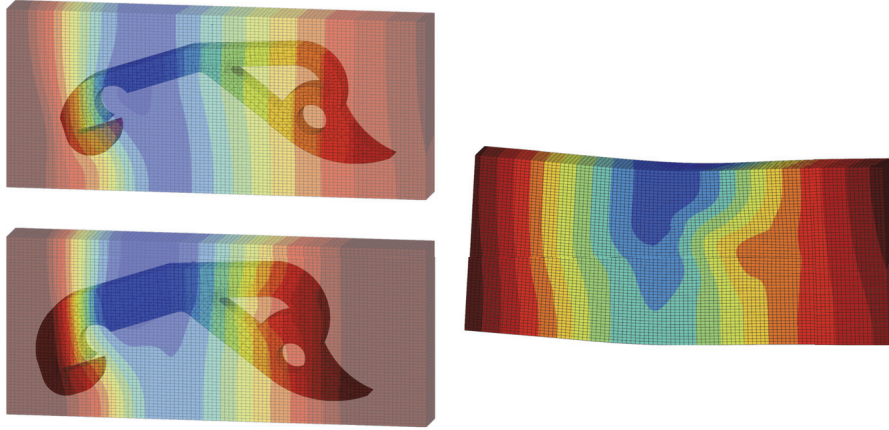


FIGURE 4: On the left, examples of vertical displacements in the bounding box superposed to the related finite element prediction, for the mesh G_1 on the top and the mesh G_4 on the bottom, with a colour scale related to the displacement magnitudes. On the right, the first feature mode $\widehat{V}[:, 1]$, for vertical displacements too.

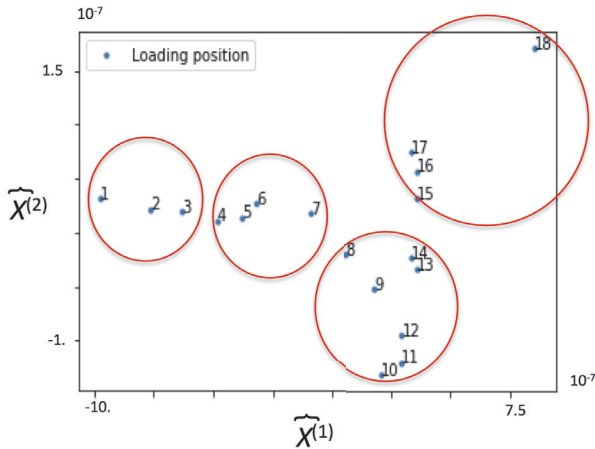


FIGURE 5: $L=18$ points $(\widehat{X}_i^{(1)}, \widehat{X}_i^{(2)})$ submitted to the k-means methods with the $K=4$ clusters shown by red circles.

simulations. As we can see in Figure 10, the elements of reduced mesh G_{R^*} are not necessarily in a continuous domain. As explained in [11], the boundary conditions at the interface between G_{R^*} and the remaining elements of G^* are similar to Dirichlet boundary conditions. They are enforced by the empirical modes of the reduced basis. Then, the mechanical coupling between the discontinuous parts of the mesh G_{R^*} is enforced by the empirical modes.

The exact error on the average stress in the region of interest is 0.1%. The map of the error estimator is shown in Figure 11. The dark grey elements in the reduced mesh (see Figure 9) are connected to the dof in set P . According to (1) by substituting the hybrid reduced basis V^{HI} for V^* , the first finite element residuals are null, for i in $\{1, \dots, \text{card}(P)\}$. Hence the error estimator $\eta(x, V^*)$ accounts partially for errors in finite element equilibrium equations. As shown

in [26], these errors explain the discrepancy between the hyperreduced prediction and the finite element prediction of the stress.

4. Conclusions

The proposed reduced order modeling is related to very huge parameter space of dimension twelve millions, mainly due to the input image of the loading environment. The accuracy of the stress prediction is satisfactory to assess the quality of the process with mechanical considerations, even if the region of interest is hidden by the experimental setup. It is also reasonably fast in order to be inserted into a manufacturing process, aiming for part-specific decisions.

The output of the proposed workflow has a high spatial resolution. This is achieved by coupling a PCA, a clustering and a convolutional neural network. A local error estimator aims to indicate the discrepancy between the output and the stress that a finite element simulation would give corresponding to the loading environment recognized by the convolutional neural network. But this error indicator does not evaluate recognition errors, neither error on the mechanical behaviour of the observed material. So, the hyperreduced order model may not be the best that the available data could give.

In this paper, the inputs of the reduced order model are nonparametric loading conditions. They are defined solely by images of the loading environment. Since digital colour images are third order tensors, we expect a possible generalization of the workflow to more complex thermomechanical loading environments and more complex variations for the geometry of the observed parts.

Here, no Big-Data is required to train the proposed reduced order modeling via computer vision. The training starts with a non-supervised machine learning by using a non-centred PCA and a clustering procedure, both on simulation data. Then, the CNN is trained on digital images

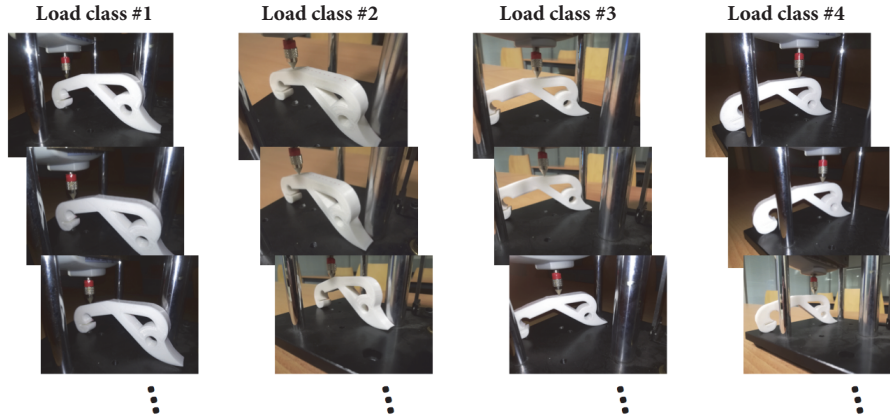


FIGURE 6: Sample images for each of the four loading environment classes.

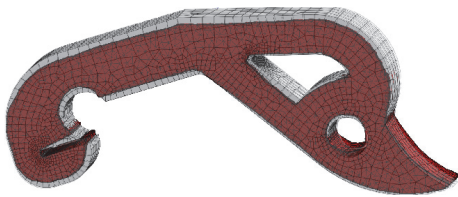


FIGURE 7: G^* is in red; the closest ideal mesh is G_4 in grey.

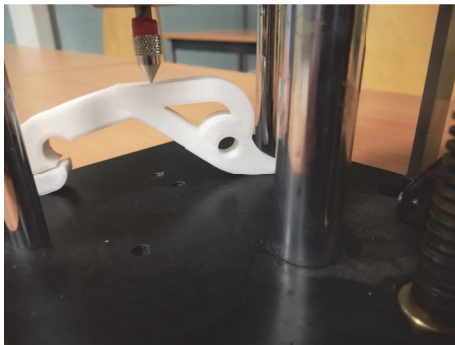


FIGURE 8: Digital image I^* of the part in the experimental setup.

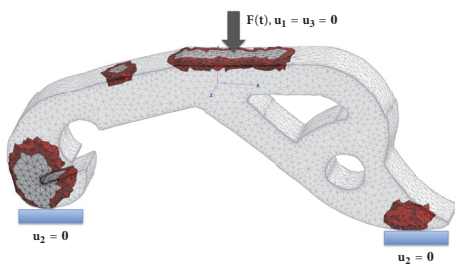


FIGURE 9: A transparency effect has been added to the full mesh G^* . Both dark grey and red finite elements are in the reduced mesh G_R^* . The grey elements are connected to degrees of freedom in the set P related to the hybrid approximation.

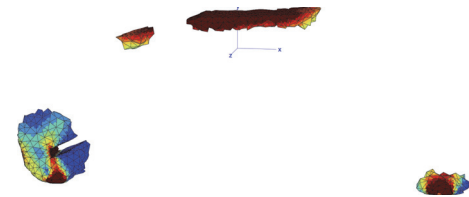


FIGURE 10: von Mises stress, $\|\sigma(x, V^* \gamma^*) - (1/3)\text{tr}[\sigma(x, V^* \gamma^*)]I\|$, in the reduced mesh G_R^* .

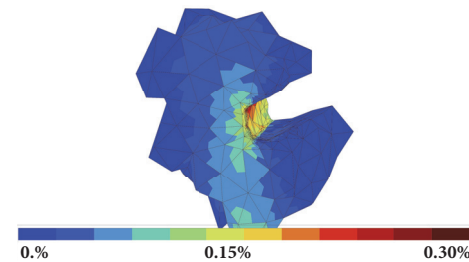


FIGURE 11: Error estimator $\eta(x, V^*) / \max_{y \in \Omega_R} \|\sigma(y, V^H \gamma^H)\| \times 100$ in the region of interest, where stresses are the highest.

by supervised machine learning upon the classes defined by the clustering procedure. Obviously, this approach can be implemented with larger sets of data.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

One of the authors is a Safran employee.

Acknowledgments

The experimental setup and the computer used for this publication have been funded by Mines ParisTech PSL Research University.

References

- [1] D. M. Pierce, W. Trobin, S. Trattnig, H. Bischof, and G. A. Holzapfel, "A Phenomenological Approach Toward Patient-Specific Computational Modeling of Articular Cartilage Including Collagen Fiber Tracking," *Journal of Biomechanical Engineering*, vol. 131, no. 9, p. 091006, 2009.
- [2] D. A. Puerto, O. C. Moreno, D. M. Gila, J. G. Garcia, and J. G. Ortega, "Olive batches automatic classification in mill reception using computer vision," in *Proceedings of the 2015 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 245–250, Macau, China, September 2015.
- [3] A. G. Stamopoulos, K. I. Tserpes, and A. J. Dentsoras, "Quality assessment of porous CFRP specimens using X-ray Computed Tomography data and Artificial Neural Networks," *Composite Structures*, vol. 192, pp. 327–335, 2018.
- [4] A. Andrieu, A. Pineau, J. Besson, D. Ryckelynck, and O. Bouaziz, "Bimodal Beremin-type model for brittle fracture of inhomogeneous ferritic steels: Theory and applications," *Engineering Fracture Mechanics*, vol. 95, pp. 84–101, 2012.
- [5] P. E. Jahromi, A. Shojaei, and S. M. Reza Pishvaie, "Prediction and optimization of cure cycle of thick fiber-reinforced composite parts using dynamic artificial neural networks," *Journal of Reinforced Plastics and Composites*, vol. 31, no. 18, pp. 1201–1215, 2012.
- [6] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 191–207, 2014.
- [7] Y. Le Cun, B. Boser, J. S. Denker et al., "Handwritten Digit Recognition with a Back-Propagation Network," in *Advances in neural information processing systems 2*, pp. 396–404, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
- [9] H. Yoo, "Deep Convolution Neural Networks in Computer Vision: a Review," *IEIE Transactions on Smart Processing and Computing*, vol. 4, no. 1, pp. 35–43, 2015.
- [10] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, "The dynamics of coherent structures in the wall region of a turbulent boundary layer," *Journal of Fluid Mechanics*, vol. 192, pp. 115–173, 1988.
- [11] D. Ryckelynck, K. Lampoh, and S. Quilicy, "Hyper-reduced predictions for lifetime assessment of elasto-plastic structures," *Meccanica*, vol. 51, no. 2, pp. 309–317, 2016.
- [12] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, "Localized discrete empirical interpolation method," *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. A168–A192, 2014.
- [13] D. Amsallem and B. Haasdonk, "PEBL-ROM: Projection-error based local reduced-order models," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, no. 6, 2016.
- [14] E. Kaiser, M. Morzyński, G. Daviller, J. N. Kutz, B. W. Brunton, and S. . Brunton, "Sparsity enabled cluster reduced-order models for control," *Journal of Computational Physics*, vol. 352, pp. 388–409, 2018.
- [15] K. Manohar, S. L. Brunton, and J. N. Kutz, "Environment identification in flight using sparse approximation of wing strain," *Journal of Fluids and Structures*, vol. 70, pp. 162–180, 2017.
- [16] F. N'Guyen, *Morphologie mathématique appliquée au développement d'outils de maillage EF automatiques dans le cas de microstructures hétérogènes bi et multiphasées [Ph.D. thesis]*, Mécanique Lille, 2014, <http://www.theses.fr/2014LIL10157>.
- [17] H. Proudhon, A. Moffat, I. Sinclair, and J.-Y. Buffiere, "Three-dimensional characterisation and modelling of small fatigue corner cracks in high strength Al-alloys," *Comptes Rendus Physique*, vol. 13, no. 3, pp. 316–327, 2012.
- [18] A. Buljac, M. Shakoor, J. Neggers et al., "Numerical validation framework for micromechanical simulations based on synchrotron 3D imaging," *Computational Mechanics*, vol. 59, no. 3, pp. 419–441, 2017.
- [19] Z. Quan, Z. Larimore, X. Qin et al., "Microstructural characterization of additively manufactured multi-directional preforms and composites via X-ray micro-computed tomography," *Composites Science and Technology*, vol. 131, pp. 48–60, 2016.
- [20] O. Zienkiewicz, R. Taylor, and J. Zhu, *Finite Element Method*, Butterworth-Heinemann, 2006.
- [21] F. Chinesta, A. Leygue, F. Bordeu et al., "PGD-Base Computational Vademecum for Efficient Design, Optimization and Control," *Archives of Computational Methods in Engineering: State-of-the-Art Reviews*, vol. 20, no. 1, pp. 31–59, 2013.
- [22] I. Oseledets and E. Tyrtshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra and its Applications*, vol. 432, no. 1, pp. 70–88, 2010.
- [23] C. Olivier, *Décompositions tensorielles et factorisations de calculs intensifs appliquées à l'identification de modèles de comportement non linéaire [Ph.D. thesis]*, Mines ParisTech, 2017, <https://pastel.archives-ouvertes.fr/tel-01783950>.
- [24] J. Baiges, R. Codina, and S. Idelsohn, "A domain decomposition strategy for reduced order models. Application to the incompressible Navier-Stokes equations," *Computer Methods Applied Mechanics and Engineering*, vol. 267, pp. 23–42, 2013.
- [25] J. Fauque, I. Ramière, and D. Ryckelynck, "Hybrid hyper-reduced modeling for contact mechanics problems," *International Journal for Numerical Methods in Engineering*, vol. 115, no. 1, pp. 117–139, 2018.
- [26] D. Ryckelynck, L. Gallimard, and S. Jules, "Estimation of the validity domain of hyper-reduction approximations in generalized standard elastoviscoplasticity," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, no. 1, 2015.
- [27] A. Badas, D. González, I. Alfaro, F. Chinesta, and E. Cueto, "Local proper generalized decomposition," *International Journal for Numerical Methods in Engineering*, vol. 112, no. 12, pp. 1715–1732, 2017.
- [28] González, J. V. Aguado, E. Cueto, E. Abisset-Chavanne, and F. Chinesta, "kPCA-Based Parametric Solutions Within the PGD Framework," *Archives of Computational Methods in Engineering: State-of-the-Art Reviews*, vol. 25, no. 1, pp. 69–86, 2018.
- [29] N. Lee, A.H. Phan, F. Cong, and A. Cichocki, "Nonnegative Tensor Train Decompositions for Multi-domain Feature Extraction and Clustering," in *Neural Information Processing. ICONIP 2016*, A. Hirose, S. Ozawa, K. Doya et al., Eds., 2016.
- [30] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2009/10.

- [31] D. C. Ciresan, U. Meier, J. Masci et al., “Flexible, high performance convolutional neural networks for image classification,” in *Proceedings of the IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, 2011.
- [32] T. C. Chu, W. F. Ranson, and M. A. Sutton, “Applications of digital-image-correlation techniques to experimental mechanics,” *Experimental Mechanics*, vol. 25, no. 3, pp. 232–244, 1985.

Research Article

Applied Koopman Theory for Partial Differential Equations and Data-Driven Modeling of Spatio-Temporal Systems

J. Nathan Kutz ¹, J. L. Proctor,² and S. L. Brunton³

¹Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

²Institute for Disease Modeling, 3150 139th Ave SE, Bellevue, WA 98005, USA

³Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA

Correspondence should be addressed to J. Nathan Kutz; kutz@uw.edu

Received 1 June 2018; Accepted 27 August 2018; Published 2 December 2018

Academic Editor: Qingdu Li

Copyright © 2018 J. Nathan Kutz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider the application of Koopman theory to nonlinear partial differential equations and data-driven spatio-temporal systems. We demonstrate that the observables chosen for constructing the Koopman operator are critical for enabling an accurate approximation to the nonlinear dynamics. If such observables can be found, then the dynamic mode decomposition (DMD) algorithm can be enacted to compute a finite-dimensional approximation of the Koopman operator, including its eigenfunctions, eigenvalues, and Koopman modes. We demonstrate simple rules of thumb for selecting a parsimonious set of observables that can greatly improve the approximation of the Koopman operator. Further, we show that the clear goal in selecting observables is to place the DMD eigenvalues on the imaginary axis, thus giving an objective function for observable selection. Judiciously chosen observables lead to physically interpretable spatio-temporal features of the complex system under consideration and provide a connection to manifold learning methods. Our method provides a valuable intermediate, yet interpretable, approximation to the Koopman operator that lies between the DMD method and the computationally intensive extended DMD (EDMD). We demonstrate the impact of observable selection, including kernel methods, and construction of the Koopman operator on several canonical nonlinear PDEs: Burgers' equation, the nonlinear Schrödinger equation, the cubic-quintic Ginzburg-Landau equation, and a reaction-diffusion system. These examples serve to highlight the most pressing and critical challenge of Koopman theory: a principled way to select appropriate observables.

1. Introduction

Data-driven mathematical methods are increasingly important for characterizing complex systems across the physical, engineering, social, and biological sciences. These methods are aimed at discovering and exploiting a relatively small subset of the full space where low-dimensional models can be used to describe the evolution of the system. Thus, solutions can often be approximated through dimensionality reduction methods where if n is the dimension of the original high-dimensional system and r is the dimension of the subspace (or slow-manifold) where the dynamics is embedded, then $r \ll n$. The reduced order modeling (ROM) community has used this to great effect in applications such as large-scale patterns of atmospheric variability [1], turbulent flow

control architectures [2], and/or spatio-temporal encodings in neurosensory systems [3]. Traditionally, the large-scale dynamics may be embedded in the low-dimensional space using, for instance, the proper orthogonal decomposition (POD) in conjunction with Galerkin projection. More recently, the *Dynamic Mode Decomposition* (DMD) and its Koopman generalization have garnered attention due to the fact that they can (i) discover low-rank spatio-temporal patterns of activity and (ii) they can embed the dynamics in the subspace in an equation-free manner, unlike the Galerkin-POD method of ROMs. In this manuscript, we demonstrate that the judicious, and parsimonious, selection of observables for the Koopman architecture can yield accurate low-dimensional embedding for nonlinear partial differential equations (PDEs) while keeping computational

costs down and avoiding costly cross-validation. Critical to its success is an appropriate choice of observables, which is demonstrated to act as a nonlinear manifold learning method. We demonstrate the success of the method, and compare it to traditional DMD, on several canonical PDE models: Burgers' equation, the nonlinear Schrödinger equation, the cubic-quintic Ginzburg-Landau equation, and a $\lambda - \omega$ reaction-diffusion system.

Historically, the DMD method originated in the fluid dynamics community as a principled technique to decompose complex flows into a simple representation based on low-rank spatio-temporal coherent structures. Schmid [4] first defined the DMD algorithm and demonstrated its ability to provide physically interpretable insights from high-dimensional fluid data. The growing success of DMD stems from the fact that it is an *equation-free* data-driven method capable of providing an accurate decomposition of a complex system into spatio-temporal coherent structures that may be used for diagnostic analysis, short-time future state prediction, and control. Importantly, Rowley et al. [5] showed that DMD is connected to the underlying nonlinear dynamics through Koopman operator theory [6] and is readily interpretable using standard dynamical system techniques [7–10]. Specifically, the DMD algorithm is a manifestation of Koopman theory when the observable functions are the identity or a linear transformation of the underlying state space. Thus, DMD is a principled algorithmic architecture allowing for an explicit approximation of the Koopman operator. For more details, there are numerous detailed references [5, 11, 12].

The approximation of the Koopman operator via DMD is critically important for enabling evaluation of the operator from data. Indeed, it transforms Koopman theory from an abstract mathematical conception to a readily tractable computation. It also highlights the important role played by observables and their associated evolution manifolds. In particular, nonlinear PDEs can be thought to evolve on manifolds which are often difficult to characterize and are rarely known analytically. A correct choice of observables can, in some cases, *linearize* the nonlinear manifold. For instance, the nonlinear evolution governed by Burgers' PDE equation can be linearized by the Cole-Hopf transformation, thus providing a linear manifold which can trivially describe the evolution dynamics. Such exact solutions to nonlinear PDEs are extremely rare and do not often exist in practice, with the inverse scattering transform (IST) for Korteweg-deVries, nonlinear Schrödinger, and other integrable PDEs being the notable exceptions [13]. Regardless, judiciously chosen observables can help transform a PDE evolving on a strongly nonlinear manifold to a weakly nonlinear manifold, enabling a more accurate and broader range of applicability of the Koopman approximation.

The selection of appropriate observables remains one of the most important and open challenges for Koopman theory. The widely used DMD algorithm simply takes the state space variable as the observable. This provides the simplest approximation to the Koopman operator. An alternative approach has been advocated by Williams et al. and Kevrekidis et al. [14, 15] whereby machine learning concepts

(e.g., support vector machines (SVM)) are used to project the data to a large number of variables using the so-called extended DMD and kernel DMD (EDMD) methods. Thus, the DMD approximation is computed in a large set of nonlinear observables of the original data. Recently, it has been shown that the EDMD method is equivalent to the *variational approach of conformation dynamics* (VAC) [16, 17], which was first derived by Noé and Nüske in 2013 for simulating slow processes in stochastic dynamical systems applied to molecular kinetics [18, 19]. The authors further show that *time-lagged independent component analysis* (TICA) [20, 21], which was originally developed in 1994, is closely related to the DMD algorithm [17]. Regardless of the EDMD/VAC strategy, it is well known in the machine learning literature, such projections into higher dimensional space, through SVM or deep neural nets, can lead to improved predictions at the cost of loss of interpretability. It also projects to variables that may have no natural association with the underlying physics or nonlinear manifold of the dynamics being considered. Importantly, Klus et al. [17] show that the EDMD/VAC method requires a principled cross-validation strategy in order to make the technique useful.

Our approach is aimed at improving the straightforward DMD approximation by adding a parsimonious set of judiciously chosen variables which are motivated by the governing equations, i.e., it is a version of EDMD with only a few extra variables. Thus, simple choices of nonlinear observables can greatly improve the Koopman approximation. Moreover, we show that a clear goal in selecting observables is to move DMD eigenvalues onto the imaginary axis. We show that selecting a parsimonious set of observables allows us to capitalize on the EDMD architecture while only incurring a marginal increase in computational costs and avoiding costly cross-validation for computing an improved approximation to the Koopman operator. Further, the judicious choice of observables can also be used to help understand the nonlinear manifold on which the dynamics evolve. Ultimately, this provides a valuable third option for variable selection that sits between the standard application of the DMD and EDMD methods.

2. Koopman Theory, Observables, and Dynamic Mode Decomposition

The original work of Koopman in 1931 [6] considered Hamiltonian systems and formulated the Koopman operator as a discrete-time mapping. In the following year, Koopman and von Neumann extended these results to dynamical systems with continuous spectra [22]. Critical to implementing this definition numerically is understanding how to choose a finite set of observables $g(x)$. This remains an open challenge today and will be addressed in our PDE examples.

By construction, the Koopman operator is a *linear* infinite-dimensional operator that acts on the Hilbert space \mathcal{H} of *all* scalar measurement functions g . The Koopman operator acts on functions of the state space of the dynamical system, trading nonlinear finite-dimensional dynamics for

linear infinite-dimensional dynamics. It can be further generalized to map infinite-dimensional nonlinear dynamics to infinite-dimensional linear dynamics by appropriate choice of observables. In practice, the computation of the Koopman operator will require a finite-dimensional representation. The advantage of the Koopman representation is compelling: linear problems can be solved using standard linear operator theory and spectral decompositions. With such methods, the infinite dimensional representation is handled by considering a sufficiently large, but finite, sum of modes to approximate the Koopman spectral solution.

The Koopman operator is defined for discrete-time dynamical systems. A continuous dynamical system will induce a discrete-time dynamical system given by the flow map $\mathbf{F}_t : \mathcal{M} \rightarrow \mathcal{M}$, which maps the state $\mathbf{x}(t_0)$ to a future time $\mathbf{x}(t_0 + t)$:

$$\mathbf{F}_t(\mathbf{x}(t_0)) = \mathbf{x}(t_0 + t) = \mathbf{x}(t_0) + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\tau)) d\tau. \quad (1)$$

This induces the discrete-time dynamical system

$$\mathbf{x}_{k+1} = \mathbf{F}_t(\mathbf{x}_k), \quad (2)$$

where $\mathbf{x}_k = \mathbf{x}(kt)$. The analogous discrete-time Koopman operator is given by \mathcal{K}_t such that $\mathcal{K}_t g = g \circ \mathbf{F}_t$. Thus, the Koopman operator sets up a discrete-time dynamical system on the observable function g :

$$\mathcal{K}_t g(\mathbf{x}_k) = g(\mathbf{F}_t(\mathbf{x}_k)) = g(\mathbf{x}_{k+1}). \quad (3)$$

If an appropriate Koopman operator can be constructed, then linear operator theory provides the spectral decomposition required to represent the dynamical solutions of interest. Specifically, the eigenfunctions and eigenvalues of the Koopman operator \mathcal{K} give a complete characterization of the dynamics. We consider the eigenvalue problem

$$\mathcal{K} \varphi_k = \lambda_k \varphi_k. \quad (4)$$

The functions $\varphi_k(\mathbf{x})$ are Koopman eigenfunctions, and they define a set of intrinsic measurement coordinates, on which it is possible to advance these measurements with a *linear* dynamical system. The low-dimensional embedding of the dynamics is ultimately extracted from the Koopman eigenfunctions. More precisely, a reduced-order linear model can be constructed by a rank- r truncation of the dominant eigenfunctions φ_k .

A vector of observables \mathbf{g} , which is in our new *measurement space*, may be expressed in terms of Koopman eigenfunctions φ as

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_p(\mathbf{x}) \end{bmatrix} = \sum_{k=1}^{\infty} \varphi_k(\mathbf{x}) \mathbf{v}_k, \quad (5)$$

where \mathbf{v}_k is the k th Koopman mode associated with the k th Koopman eigenfunction φ_k , i.e., it is the weighting of each observable on the eigenfunction. In the original theory [6], Koopman considered Hamiltonian flows that are measure preserving, so that the Koopman operator is unitary. In this case, the eigenfunctions are all orthonormal, and (5) may be written explicitly as

$$\mathbf{g}(\mathbf{x}) = \sum_{k=1}^{\infty} \varphi_k(\mathbf{x}) \begin{bmatrix} \langle \varphi_k, g_1 \rangle \\ \langle \varphi_k, g_2 \rangle \\ \vdots \\ \langle \varphi_k, g_p \rangle \end{bmatrix} = \sum_{k=1}^{\infty} \varphi_k(\mathbf{x}) \mathbf{v}_k. \quad (6)$$

The dynamic mode decomposition algorithm is used to compute an approximation to the Koopman eigenvalues λ_k and modes \mathbf{v}_k .

The nonlinear dynamical system defined by f and the infinite-dimensional linear dynamics defined by \mathcal{K} are equivalent representations of a dynamical system. One can either evolve the system in the original state space, requiring computational effort since it is nonlinear, or one can instead evolve using (5) so that the time dynamics are trivially computed

$$\mathcal{K} g(\mathbf{x}) = \mathcal{K} \sum_{k=1}^{\infty} \varphi_k(\mathbf{x}) \mathbf{v}_k = \sum_{k=1}^{\infty} \mathcal{K} \varphi_k(\mathbf{x}) \mathbf{v}_k = \sum_{k=1}^{\infty} \lambda_k \varphi_k(\mathbf{x}) \mathbf{v}_k. \quad (7)$$

Thus, future solutions can be computed by simple multiplication with the Koopman eigenvalue. Such a mathematical strategy for evolving nonlinear dynamical systems would always seem to be advantageous. However, it remains an open challenge how to systematically link the observables g and the associated Koopman mode expansion to the original evolution defined by f . For a limited class of nonlinear dynamics, this can be done explicitly [23].

In theory, the modification of the Koopman operator to PDEs would generate eigenfunctionals of the Koopman operator. In practice, the discretization of space and time, either in experiment or simulation, yields a high-dimensional system of ODEs. The PDE itself imposes clear relations between the high-dimensional data which correspond to spatial locations. Specifically, the data generated from the PDE most certainly inherit the underlying dynamics enforced by spatial relations and their spatial derivatives, leading to dimensionality-reduction and low-rank truncation possibilities. In our examples, the spatio-temporal patterns can be represented in POD/DMD modes with a truncation using the dominant r modes. This truncation to a low-dimensional space is a direct consequence of the PDE nature of the solutions. For a thorough discussion of the difference simply between high-dimensional ODEs and PDEs, please see [24].

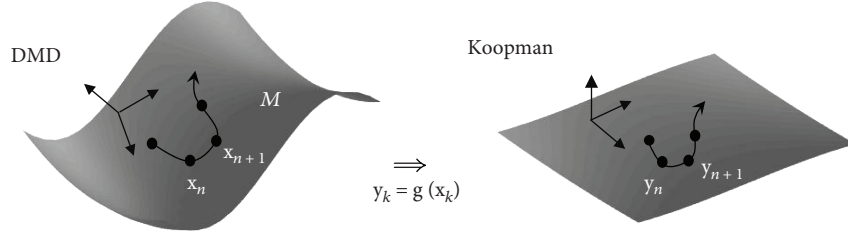


FIGURE 1: The left panel illustrates the nonlinear manifold on which the dynamical system defined by F_t in (2) generates a solution. DMD approximates the evolution on this manifold by a least-square fit linear dynamical system. In contrast, the selection of appropriate observables $g(x)$ define a Koopman operator that helps *linearize* the manifold so that a least-square fit linear dynamical system provides a much better approximation to the system (right panel).

Figure 1 illustrates the underlying concept in the Koopman approach. A dynamical system consisting of snapshots x_k evolves according to the nonlinear dynamical system defined by F_t in (2). In the state space, the nonlinearity generates a nonlinear manifold in which the data are embedded. The DMD approximation produces a least-square fit *linear dynamical system* approximating the flow map and the low-dimensional embedding (left panel of Figure 1). Koopman theory ideally defines an operator that attempts to *linearize* the space in which the data are embedded. The Koopman operator then produces a linear flow map and low-dimensional embedding that approximates the full nonlinear dynamics (right panel of Figure 1).

The nonlinear manifold on which the dynamics evolve can change due to parameter changes in the PDE. For instance, the dynamics can undergo bifurcation and generate a new nonlinear manifold. This requires building a new Koopman operator to characterize the dynamics. The Koopman embedding can be used to build libraries of low-rank representations for the dynamics. The concept of library building of low-rank “features” from data is well established in the computer science community. In the reduced-order modeling community, it has recently become an issue of intense investigation. Indeed, a variety of recent works have produced libraries of ROM models that can be selected and/or interpolated through measurement and classification [25–31]. Alternatively, cluster-based reduced order models use a k-means clustering to build a Markov transition model between dynamical states [32]. More recently, such techniques have been applied using the DMD approximation for the Koopman operator [33]. The modeling of parametric systems remains an open challenge for model reduction frameworks.

3. The DMD and Koopman Algorithms

The DMD algorithm underlies the computation of the Koopman eigenvalues and modes directly from data. Its effectiveness depends sensitively on the choice of observables. Rowley et al. [5] showed that DMD approximates the Koopman operator for the set of observables $g(x) = x$. We will use this fact in constructing a DMD algorithm for observables of x instead of the state variable itself. To start, we use the following definition of the DMD decomposition [11]:

Definition 1. Dynamic mode decomposition [11]: suppose we have a dynamical system (2) and two sets of data

$$X = \begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_m \\ | & | & & | \end{bmatrix}, \quad (8a)$$

$$X' = \begin{bmatrix} | & | & \cdots & | \\ x'_1 & x'_2 & \cdots & x'_m \\ | & | & & | \end{bmatrix}, \quad (8b)$$

with x_k an initial condition to (2) and x'_k its corresponding output after some prescribed evolution time Δt with there being m initial conditions considered. The DMD modes are eigenvectors of

$$A_X = X'X^\dagger, \quad (9)$$

where \dagger denotes the Moore-Penrose pseudoinverse.

The above definition provides a computational method for evaluating the Koopman operator for a linear observable. In practice, three practical constraints must be considered: (i) We have data X and X' , but we do not necessarily know $F_t(\cdot)$, (ii) We will have to make a finite-dimensional approximation to the infinite-dimensional Koopman operator \mathcal{K} , and (iii) We will have to judiciously select the observables $g(x)$ in order to have confidence that that Koopman operator will approximate the nonlinear dynamics of $F_t(\cdot)$. Points (i) and (ii) go naturally together. Specifically, the number of measurements in each column of X and X' is n , while the number of total columns (time measurements) is m . Thus, finite-dimensionality is imposed simply from the data collection limitations. The dimension can be increased with a large set of observables, or it can be decreased via a low-rank truncation during the DMD process. The observables are more difficult to deal with in a principled way. Indeed, a good choice of observables can make the method extremely effective, but it would also require expert knowledge of the system at hand [23]. This will be discussed further in the examples.

The following gives a practical demonstration of how to use the data, the DMD algorithm, and the observables to produce a Koopman operator and a future state prediction of the nonlinear evolution. The Koopman algorithm simply applies DMD on the space of observables.

- (1) From the data matrices \mathbf{X} and \mathbf{X}' , create the data matrices of observables \mathbf{Y} and \mathbf{Y}' :

$$\mathbf{Y} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{g}(x_1) & \mathbf{g}(x_2) & \cdots & \mathbf{g}(x_{m-1}) \\ | & | & & | \end{bmatrix}, \quad (10a)$$

$$\mathbf{Y}' = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{g}(x'_1) & \mathbf{g}(x'_2) & \cdots & \mathbf{g}(x'_{m-1}) \\ | & | & & | \end{bmatrix}, \quad (10b)$$

where each column is given by $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k)$ or $\mathbf{y}'_k = \mathbf{g}(\mathbf{x}'_k)$

- (2) Perform the DMD algorithm on the pair \mathbf{Y} and \mathbf{Y}' to compute

$$\mathbf{A}_Y = \mathbf{Y}'\mathbf{Y}^\dagger \quad (11)$$

along with the low-rank counterpart $\tilde{\mathbf{A}}_Y$ obtained by projection onto a truncated POD subspace. The eigenvalues and eigenvectors of \mathbf{A}_Y may approximate Koopman eigenvalues and modes if the observables are well chosen

- (3) DMD can be used to compute the augmented modes Φ_Y , which may approximate the Koopman modes, by

$$\Phi_Y = \mathbf{Y}'\mathbf{V}\Sigma^{-1}\mathbf{W}, \quad (12)$$

where \mathbf{W} comes from the eigenvalue problem $\tilde{\mathbf{A}}_Y\mathbf{W} = \mathbf{W}\Lambda$ and $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^*$. Note that an r -rank truncation of the SVD is performed at this stage

- (4) The future state in the space of observables is given by the linear evolution

$$\mathbf{y}(t) = \Phi_Y \text{diag}(\exp(\omega t)) \mathbf{b}, \quad (13)$$

where $\mathbf{b} = \Phi_Y^\dagger \mathbf{y}_1$ is determined by projecting back to the initial data observable. The continuous-time eigenvalues ω are obtained from the discrete-time eigenvalues λ_k (i.e., diagonal elements of the matrix Λ) where $\omega_k = \ln(\lambda_k)/\Delta t$

- (5) Transform from observables to state space

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) \longrightarrow x_k = \mathbf{g}^{-1}(\mathbf{y}_k). \quad (14)$$

This last step is trivial if one of the observables selected to comprise $\mathbf{g}(\mathbf{x}_k)$ is the state variable x_k itself. If only nonlinear observables of x_k are chosen, then the inversion process can be difficult.

This process shows that the DMD algorithm is closely related to the Koopman operator. Indeed, it is the foundational piece for practical evaluation of the finite-dimensional Koopman operator. It is stressed once again here: selection of appropriate observables is critical for the algorithm to generate good reconstructions and approximations to the future state. We can also now introduce the following theorem [5, 11, 14, 34].

Theorem 1. *Koopman and dynamic mode decomposition: let φ_k be an eigenfunction of \mathcal{X} with eigenvalue λ_k , and suppose $\varphi_k \in \text{span}\{g_j\}$, so that*

$$\varphi_k(\mathbf{x}) = w_1 g_1(\mathbf{x}) + w_2 g_2(\mathbf{x}) + \cdots + w_p g_p(\mathbf{x}) = \mathbf{w} \cdot \mathbf{g} \quad (15)$$

for some $\mathbf{w} = [w_1 w_2 \cdots w_p]^T \in \mathbb{C}^p$. If $\mathbf{w} \in R(\mathbf{Y})$, where R is the range, then \mathbf{w} is a left eigenvector of \mathbf{A}_Y with eigenvalue λ_k so that $\tilde{\mathbf{w}}^* \mathbf{A}_Y = \lambda_k \tilde{\mathbf{w}}^*$.

Note here that the observables $g_j(x)$ as introduced in the theorem [5, 11, 14, 34] are not denoted as vectors since the theorem applies to functions. However, in practice, when a system is discretized, then $g_j(x) \longrightarrow g_j(x)$ as is explicitly constructed in the algorithm above. Thus, the Koopman eigenvalues are the DMD eigenvalues provided (i) the set of observables is sufficiently large so that $\varphi_k(x) \in \text{span}\{g_j\}$ and (ii) the data is sufficiently *rich* so that $w \in R(X)$. This directly shows that the choice of observables is critical in allowing one to connect DMD theory to Koopman spectral analysis. If this can be done, then one can simply take data snapshots of a finite-dimensional nonlinear dynamical system in time and reparameterize it as a linear system in the observable coordinates, which is amenable to a simple eigenfunction (spectral) decomposition. This representation diagonalizes the dynamics and shows that the time evolution of each eigenfunction corresponds to multiplication by its corresponding eigenvalue.

4. Koopman Observables and Kernel Methods

The effectiveness of Koopman theory hinges on one thing: selecting appropriate observables. Once observables are selected, the previous section defines a DMD-based algorithm for computing the Koopman operator whose spectral decomposition completely characterizes the approximation. In the machine learning literature, observables are often the basis of generating *features*, and we will build upon this concept to generate appropriate observables. An important practical consideration becomes the computational cost in generating the DMD approximation as the number of rows in the matrices \mathbf{Y} and \mathbf{Y}' gets progressively larger with each additional observable.

To be more precise about the distinction between the EDMD/VAC and the present work, we emphasize that the state-of-the-art machine learning approach to EDMD is given by Williams et al. [14] and Klus et al. [17]. These works provide a framework for projection to observables through kernel SVM-like methods, as well as showing how to cross-validate the results. As is common in such schemes, interpretability and generalizability is typically lost. In the current work, our goal is to select a parsimonious set of observables based upon knowledge of the physics or its constraints. It provides a viable strategy whereby expert knowledge can be leveraged to guide the selection of observables. In the previous works [14, 17], parsimony was not used in the regression framework. Here, parsimony to produce interpretable models, and potentially generalizable models, is the critical innovation advocated. It should be noted that we illustrate some of the kernel methods here in order to simply show the lack of robustness of EDMD in the absence of parameter tuning.

In the absence of expert-in-the-loop knowledge of the dynamical system, one might consider, for instance, the support vector machine (SVM) literature and associated kernel methods [35–38] for feature selection (observables). The SVM architecture suggests a number of techniques for constructing the feature space $g(x)$, with a common choice being the set of polynomials such that

$$g_j(x) = \{x, x^2, x^3, x^4, \dots, x^p\}. \quad (16)$$

Using a large number of polynomials can generate an extremely large vector of observables for each snapshot in time. This is closely related to the Carleman linearization technique in dynamical systems [39–41]. Alternatively, kernel methods have found a high degree of success using (i) radial basis functions, typically for problems defined on irregular domains, (ii) Hermite polynomials for problems defined on \mathbb{R}^n , and (iii) discontinuous spectral elements for large problems with block diagonal structures. Regardless of the specific choice of feature space, the goal is to choose a sufficiently rich and diverse set of observables that allow an accurate approximation of the Koopman operator \mathcal{K} . Instead of choosing the correct observables, one then simply chooses a large set of candidate observables with the expectation that a sufficiently diverse set will include enough features for an accurate reconstruction of the Koopman modes, eigenfunctions, and eigenvalues, which intrinsically characterize the nonlinear dynamical system.

Williams et al. and Kevrekidis et al. [14, 15] have recently capitalized on the ideas of machine learning by implementing the so-called extended DMD and kernel DMD method on extended observables (16) within the DMD architecture. Moreover, they have developed an efficient way to compute $\tilde{\mathbf{A}}_Y$ even for a large observable space. The kernel DMD method is the most relevant in practice as the number of observables (features) can rapidly grow so as to make n extremely high-dimensional. In the context of the Koopman operator, the kernel trick [35–38] will define a function $h(x, x')$ that can be related to the observables $g_j(x)$ used

for constructing \mathbf{Y} and \mathbf{Y}' . Consider the simple example of a quadratic polynomial kernel

$$h(x, x') = (1 + x^T x')^2, \quad (17)$$

where x and x' are data points in \mathbb{R}^2 . When expanded, the kernel function takes the form

$$\begin{aligned} h(\mathbf{x}, \mathbf{x}') &= (1 + x_1 x_1' + x_2 x_2')^2 \\ &= (1 + 2x_1 x_1' + 2x_2 x_2' + 2x_1 x_2 x_1' x_2' + x_1^2 x_1'^2 + x_2^2 x_2'^2) \\ &= 1 + \mathbf{Y}^T(\mathbf{x}') \mathbf{Y}(\mathbf{x}), \end{aligned} \quad (18)$$

where $\mathbf{Y}(\mathbf{x}) = [\sqrt{2}x_1 \quad \sqrt{2}x_2 \quad \sqrt{2}x_1 x_2 \quad x_1^2 \quad x_2^2]^T$. Note that for this case, both the Koopman observables and the kernel function (17) are equivalent representations that are paired together through the expansion (18). The so-called kernel trick posits that (17) is a significantly more efficient representation of the polynomial variables that emerge from the expansion (18). Instead of defining the Koopman observables $g_i(x)$, we instead define the kernel function (17) as it provides a compact representation of the infinite-dimensional feature space and an implicit computation of the inner products required for the Koopman operator.

The computational advantages of the kernel trick are considerable. For example, a polynomial kernel of degree p acting on data vectors \mathbf{x} and \mathbf{x}' in \mathbb{R}^n is given by

$$h(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^p, \quad (19)$$

which requires a single computation of the inner product $\alpha = \mathbf{x}^T \mathbf{x}'$. This requires $\mathcal{O}(n)$ and produces $f(x, x') = (1 + \alpha)^p$ where α is a constant. The resulting computational cost for this p th degree polynomial kernel remains $\mathcal{O}(n)$. In contrast, the alternative observable space using Y requires construction of a vector of length

$$O\binom{n+p}{p} \quad (20)$$

taking the form

$$\mathbf{Y}(\mathbf{x}) = [1 \quad x_1 \quad \dots \quad x_n \quad x_1^2 \quad x_1 x_2 \quad \dots \quad x_n^2 \quad \dots \quad x_1^p \quad \dots \quad x_n^p]^T. \quad (21)$$

Computing the inner product $\mathbf{Y}^T(\mathbf{x}') \mathbf{Y}(\mathbf{x})$ is a significantly larger computation than the kernel form (19). Thus, the kernel trick enables an advantageous representation of the various polynomial terms and circumvents the formation and computation associated with (21). It should be noted that these two representations are not equivalent. Rather, they give two different representations of nonlinear observables from which a feature space can be extracted. In the former

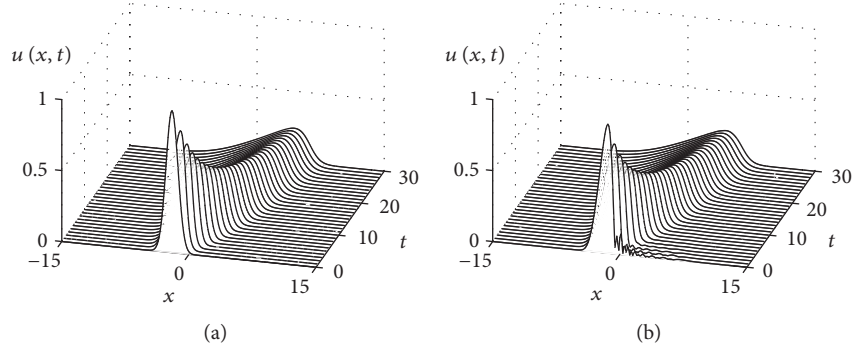


FIGURE 2: (a) Evolution dynamics of Burgers' equation with initial condition $u(x, 0) = \exp(-(x+2)^2)$. (b) Fifteen mode DMD approximation of the Burgers' evolution. The simulation of (24) was performed over $t \in [0, 30]$ where the sampling was taken at every $\Delta t = 1$. The domain was discretized with $n = 256$ points on a domain $x \in [-15, 15]$.

(19), the computations are tractable, while in the latter (21), the computation quickly becomes intractable.

The choice of kernel is important and in practice is not robust for Koopman methods. Some standard choices are often used, including the three most common kernels of SVM-based data methods:

$$\text{polynomial kernel (degree } p) h(\mathbf{x}, \mathbf{x}') = (a + \mathbf{x}^T \mathbf{x}')^p, \quad (22a)$$

$$\text{radial basis functions } h(\mathbf{x}, \mathbf{x}') = \exp(-a|\mathbf{x} - \mathbf{x}'|^2), \quad (22b)$$

$$\text{sigmoid kernel } h(\mathbf{x}, \mathbf{x}') = \tanh(\mathbf{x}^T \mathbf{x}' + a). \quad (22c)$$

The advantage of the kernel trick is quite clear, providing a compact representation of a very large feature space. For the polynomial kernel, for instance, a 20th-degree polynomial ($p=20$) using (22a) is trivial and does not compute all the inner products directly. In contrast, using our standard Koopman observables $g(x_j)$ would require one to explicitly write out all the terms generated from a 20th-degree polynomial on an n -dimensional data set, which is computationally intractable for even moderately large n . The tuning parameter a must be carefully chosen in practice for reasonable results.

In practice, the observables for \mathbf{Y} are implicitly embedded in the kernel $h(x, x')$. Specifically, we consider the observable matrix elements defined by

$$\mathbf{Y}^T \mathbf{Y}'(j, k) = h(\mathbf{x}_j, \mathbf{x}'_k), \quad (23)$$

where (j, k) denotes the j th row and k th column of the correlation matrix, and x_j and x'_k are the j th and k th columns of data. The kernel DMD formulation still requires the computation of the matrices \mathbf{V} and $\mathbf{\Sigma}$ which can be produced from $\mathbf{Y}^* \mathbf{Y} \mathbf{V} = \mathbf{\Sigma}^2 \mathbf{V}$. As before, the matrix elements of $\mathbf{Y}^* \mathbf{Y}$ are computed from $\mathbf{Y}^* \mathbf{Y}(j, k) = h(x_j, x_k)$. Thus, all the required inner products are computed by projecting directly to the new feature space defined by the specific kernel used. Note

that if the linear kernel function $h(x, y) = x^T y$ is chosen, the kernel DMD reduces to the standard DMD algorithm.

5. Application to PDEs

To demonstrate the Koopman operator concepts, we apply the methodology to various illustrative and canonical PDEs: Burgers' equation, nonlinear Schrödinger equation, the cubic-quintic Ginzburg-Landau equation, and a reaction-diffusion model. With these examples, we can (i) illustrate a scenario where the Koopman operator can exactly (analytically) linearize a dynamical system, (ii) demonstrate how to judiciously select observables, and (iii) show that kernel methods are highly sensitive as an observable selection technique.

The simulations are based upon a pseudospectral technique whereby the spatial domain and its derivatives are computed in the Fourier domain using the Fast Fourier Transform (FFT). The time-stepping algorithm is based upon an adaptive 4th-order Runge-Kutta scheme, i.e., *ode45* in MATLAB. By default, the FFT-based strategy imposes periodic boundary conditions.

5.1. Burgers' Equation. To demonstrate the construction of a specific and exact Koopman operator, we consider the canonical nonlinear PDE: Burgers' equation with diffusive regularization. The evolution, as illustrated in Figure 2(a), is governed by diffusion with a nonlinear advection [42]:

$$u_t + uu_x - \epsilon u_{xx} = 0 \quad \epsilon > 0, \quad x \in [-\infty, \infty]. \quad (24)$$

When $\epsilon = 0$, the evolution can lead to shock formation in finite time. The presence of the diffusion term regularizes the PDE, ensuring continuous solutions for all time.

Burgers' equation is one of the few nonlinear PDEs whose analytic solution form can be derived. In independent seminal contributions, Hopf [43] and Cole [44] derived a transformation that linearizes the PDE. The Cole-Hopf transformation is defined as follows

$$u = -2\epsilon v_x / v. \quad (25)$$

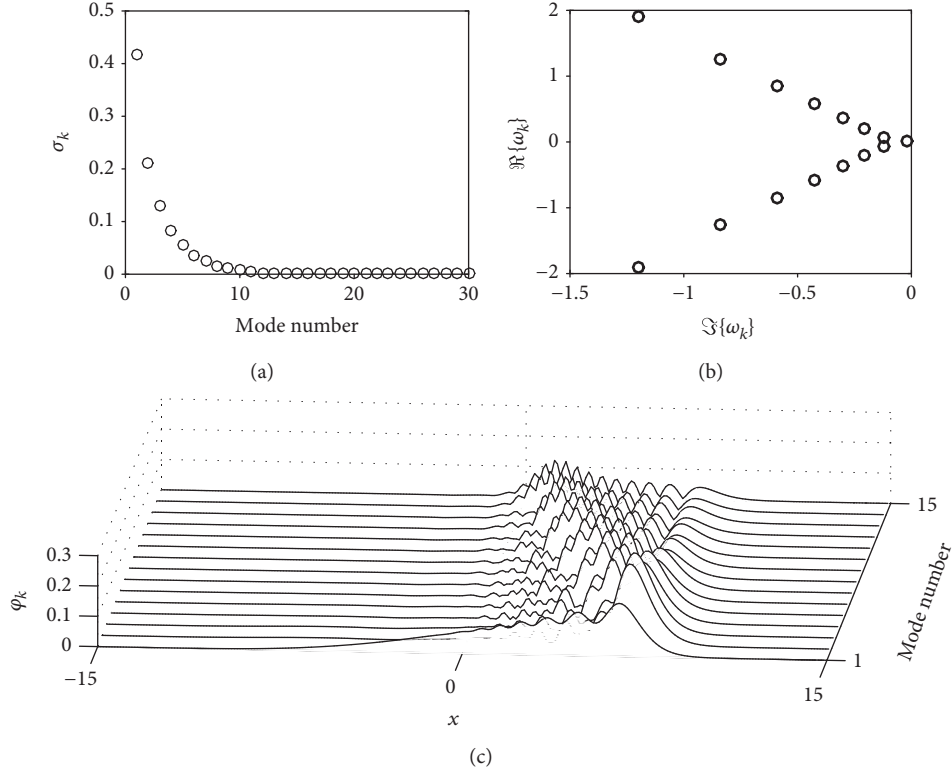


FIGURE 3: DMD of the Burgers' equation. (a) The singular value spectrum demonstrates that a rank $r = 15$ truncation should be adequate to capture the dynamics of the front propagation in Figure 2. (b) The eigendecomposition in the DMD algorithm produces a DMD spectra whose eigenvalues are decaying. (c) The DMD modes used for reconstructing the solution in Figure 2 ordered according to the smallest (in magnitude) eigenvalues. The first mode is like a *background* mode since the eigenvalue is almost zero.

The transformation to the new variable $v(x, t)$ replaces the nonlinear PDE (24) with the linear diffusion equation

$$v_t = \epsilon v_{xx}, \quad (26)$$

where it is noted that $\epsilon > 0$ in (24) in order to produce a well-posed PDE.

The diffusion equation can be easily solved using Fourier transforms. Fourier transforming in x gives the ODE system

$$\hat{v}_t = -\epsilon k^2 \hat{v}, \quad (27)$$

where $\hat{v} = \hat{v}(k, t)$ denotes the Fourier transform of $v(x, t)$ and k is the wavenumber. The solution in the Fourier domain is easily found to be

$$\hat{v} = \hat{v}_0 \exp(-\epsilon k^2 t), \quad (28)$$

where $\hat{v}_0 = \hat{v}(k, 0)$ is the Fourier transform of the initial condition $v(x, 0)$.

To construct the Koopman operator, we can then combine the transform to the variable $v(x, t)$ from (25)

$$v(x, t) = \exp \left[-\frac{\int_{-\infty}^x u(\xi, t) d\xi}{2\epsilon} \right], \quad (29)$$

with the Fourier transform to define the observables

$$g(u) = \hat{v}. \quad (30)$$

The Koopman operator is then constructed from (28) so that

$$\mathcal{K} = \exp(-\epsilon k^2 t). \quad (31)$$

This is one of the rare instances where an explicit expression for the Koopman operator and the observables can be constructed analytically. The inverse scattering transform [13] for other canonical PDEs, Korteweg-deVries (KdV) and nonlinear Schrödinger (NLS) equations, also can lead to an explicit expression for the Koopman operator, but the scattering transform and its inversion are much more difficult to construct in practice.

To make comparison between Koopman theory and DMD, we consider the DMD method applied to governing (24). Applying the algorithm of Section 3 to the observables $g(x) = x$ gives the DMD approximation to the Burgers' dynamics as shown in Figure 2(b). For this simulation, data snapshots were collected at intervals of $\Delta t = 1$ for the time range $t \in [0, 30]$. The singular value decay for the dynamics is shown in Figure 3(a), suggesting that a rank $r = 15$ truncation is appropriate. The DMD spectra and DMD modes are illustrated in Figures 3(b) and 3(c), respectively. Thus, using

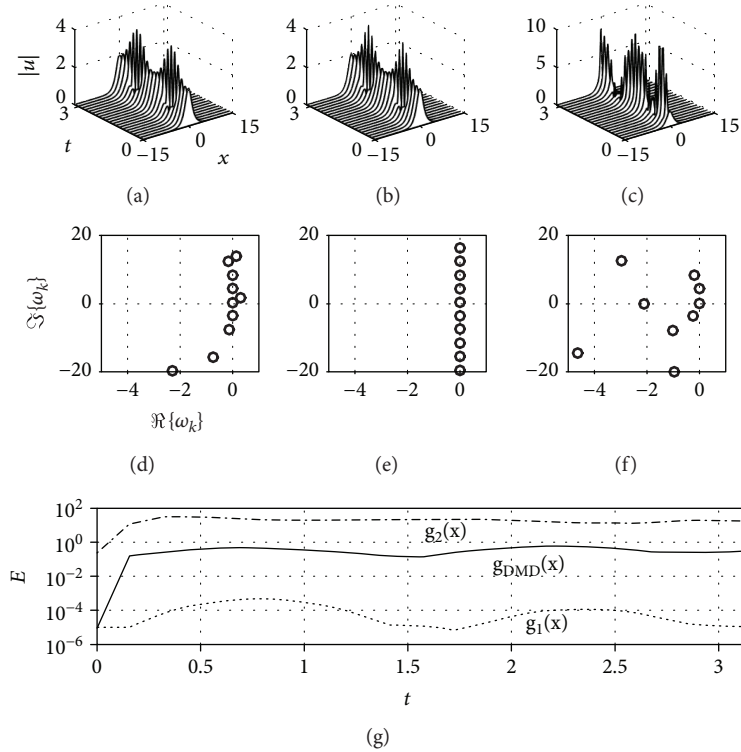


FIGURE 4: Reconstruction of the NLS dynamics using (a) a standard DMD approximation $g_{\text{DMD}}(x)$, (b) the NLS motivated $g_1(x)$, and (c) a quadratic observable $g_2(x)$. The Koopman spectra for each observable are demonstrated in (d), (e), and (f) which accompany the observables of (a), (b), and (c), respectively. Note that the observable $g_1(x)$ produces spectra which are approximately purely imaginary which is expected of the 2-soliton evolution. The error between the three observables and the full simulation is shown (g). Note that the observable $g_1(x)$ gives an error reduction of four-orders of magnitude over DMD, while $g_2(x)$ is an order of magnitude worse. This highlights the importance of selecting good observables. The simulation was performed over $t \in [0, \pi]$ with 41 equally spaced snapshots taken. The domain was discretized with $n = 512$ points on a domain $x \in [-15, 15]$.

$u(x, t)$ directly as an observable produces a low-rank model with fifteen modes. In contrast, by working with the observable (30), the Koopman operator can be trivially computed (31) and the dynamics analytically produced without need of approximation. In this case, the Koopman operator exactly linearizes the dynamics. This is the ideal which is hoped for but rarely achieved with nonlinear PDEs (or nonlinear dynamical systems in general).

5.2. Nonlinear Schrödinger Equation. The example of Burgers' equation was easy to quantify and understand since the Cole-Hopf transformation was discovered nearly seven decades ago. Thus, the observables chosen were easily motivated from knowledge of the analytic solution. Unfortunately, it is rarely the case that such linearizing transformations are known. In our second example, we consider the Koopman operator applied to a second canonical nonlinear PDE: the nonlinear Schrödinger equation

$$iu_t + \frac{1}{2}u_{xx} + |u|^2u = 0, \quad (32)$$

where $u(x, t)$ is a function of space and time modeling slowly varying optical fields or deep water waves, for instance. Discretizing in the spatial variable x , we can Fourier transform the solution in space and use a standard time-stepping

algorithm, such as a fourth-order Runge-Kutta, to integrate the solution forward in time.

As with Burgers' equation, we can compute the DMD by collecting snapshots of the dynamics over a specified time window. Specifically, we consider simulations of the equation with initial data

$$u(x, 0) = 2 \operatorname{sech}(x) \quad (33)$$

over the time interval $t \in [0, \pi]$. Twenty-one snapshots of the dynamics are collected during the evolution, allowing us to create the snapshot matrix \mathbf{X} and \mathbf{X}' . The DMD reconstruction of the dynamics is demonstrated in Figure 4(a). The low-rank DMD reconstruction provides a good approximation to the dynamics of the PDE.

To be more precise, it is explicitly assumed in the DMD reduction that the observables are simply the state variables x where $x = u(x, t)$ at discrete space and time points. The DMD observables are then given by

$$\mathbf{g}_{\text{DMD}}(\mathbf{x}) = \mathbf{x}. \quad (34)$$

Thus, as previously noted, the DMD approximation is a special case of Koopman. The DMD spectrum for a rank $r = 10$ approximation is shown in Figure 4(d). An ideal

approximation would have the eigenvalues aligned along the imaginary axis since the evolution with the initial condition given by (33) is known as the 2-soliton solution which is purely oscillatory.

Koopman theory allows us a much broader set of observables. In what follows, we consider two additional observables

$$\mathbf{g}_1(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ |\mathbf{x}|^2 \mathbf{x} \end{bmatrix}, \quad (35a)$$

$$\mathbf{g}_2(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ |\mathbf{x}|^2 \end{bmatrix} \quad (35b)$$

The first observable $g_1(x)$ is motivated by the form of the nonlinearity in the NLS equation. The second, $g_2(x)$, is chosen to have a simple quadratic nonlinearity. It has no special relationship to the governing equations. Note that the choice of the observable $|\mathbf{x}|^2$ in $g_2(x)$ is relatively arbitrary. For instance, one could consider instead $|\mathbf{x}|^5 \mathbf{x}$, x^2 , x^3 , or x^5 . These all produce similar results to $g_2(x)$ selected in (35b). Specifically, the observable $g_2(x)$ is inferior to either the DMD or judiciously selected $g_1(x)$ for the Koopman reconstruction.

As has been repeatedly stated, the success of the Koopman decomposition relies almost exclusively on the choice of observables. To demonstrate this in practice, we compute the Koopman decomposition of the NLS (32) using the two observables (35a) and (35b). The required data matrices have $2n$ rows of data, and only the state variables need to be recovered at the end of the procedure. Note that the algorithm produces both a state approximation since the first n components are actually the state vector x and approximations to the nonlinearity. The Koopman eigenfunctions and eigenvalues provide information about the evolution on the observable space.

Figures 4(b) and 4(d) show the Koopman reconstruction of the simulated data for the observables (35a) and (35b). The observable $g_1(x)$ provides an exceptional approximation to the evolution while $g_2(x)$ is quite poor. Indeed, the error of the DMD approximation and two nonlinear observables (35a) and (35b) are shown in Figure 4(g) where the following error metric is used:

$$E(t_k) = \|\mathbf{x}(t_k) - \tilde{\mathbf{x}}(t_k)\| \quad k = 1, 2, \dots, m, \quad (36)$$

where x is the full simulation and \tilde{x} is the DMD or Koopman approximation. With the choice of observable $g_1(x)$, which was judiciously chosen to match the nonlinearity of the NLS, the Koopman approximation of the dynamics is four-orders of magnitude better than a DMD approximation. A poor choice of observables, given by $g_2(x)$, gives the worse performance of all, an order of magnitude worse than DMD. Note also the difference in the Koopman spectra as shown in Figures 4(d)–4(f). In particular, note that the judicious observable $g_1(x)$ aligns the eigenvalues along the imaginary axis as is expected from the dynamics. It further

suggests that much better long-time predictions can be achieved with the Koopman decomposition using $g_1(x)$.

Observable selection in this case was facilitated by knowledge of the governing equations. However, in many cases, no such expert knowledge is available, and we must rely on data. The kernel DMD method allows one to use the kernel trick to consider a vast range of potential observables. As already highlighted, the kernel method allows for an efficient method to consider a large class of potential observables without making the observation vector $g(x)$ computationally intractable. For instance, one can consider a radial basis function kernel

$$f(\mathbf{x}, \mathbf{x}') = \exp\left(-|\mathbf{x} - \mathbf{x}'|^2\right). \quad (37)$$

The absolute value is conjectured to be important for the case of the NLS equation considered due to the nonlinear evolution of the phase. This radial basis-type function is one of the more commonly considered kernels. Other kernels that we might consider include the three following observables

$$f(\mathbf{x}, \mathbf{x}') = \left(1 + \mathbf{x}^T \mathbf{x}'\right)^{20}, \quad (38a)$$

$$f(\mathbf{x}, \mathbf{x}') = \left(1 + |\mathbf{x}^T| |\mathbf{x}'|\right)^{20}, \quad (38b)$$

$$f(\mathbf{x}, \mathbf{x}') = \exp\left(-\mathbf{x}^T \mathbf{x}'\right). \quad (38c)$$

The first function is the standard polynomial kernel of 20th degree. The second instead takes the absolute value of the variable in a polynomial in order to remove the phase, and the third is a Gaussian-type kernel that uses the same inner product as the polynomial kernel. Note that it uses the same inner product in state space but a completely different inner product in feature space. The selection of these kernels is motivated by well-known and often used kernels for real valued data. Other kernels can easily be considered. Our objective is not so much to evaluate a specific kernel but to demonstrate that kernel selection produces highly variable results so that kernel tuning via cross validation is of critical importance. It is well known that SVM and deep neural nets require significant cross-validation in order to work well. Moreover, note that complex data are rarely considered for kernel selection, so there is ambiguity about what impact this may have. Indeed, this remains an open research question in its own right. However, the NLS has a specific form of nonlinearity which is phase-independent, thus motivating some of our choices of potential kernels.

These three new kernels are compared to each other and the radial basis function. Figure 5 shows the spectra generated by these four kernels along with a comparison to the Koopman spectra generated by $g_1(x)$. Note the tremendous variability of the results based upon the choice of kernel. More precisely, it simply highlights the critical importance of calibrating the kernel through cross-validation.

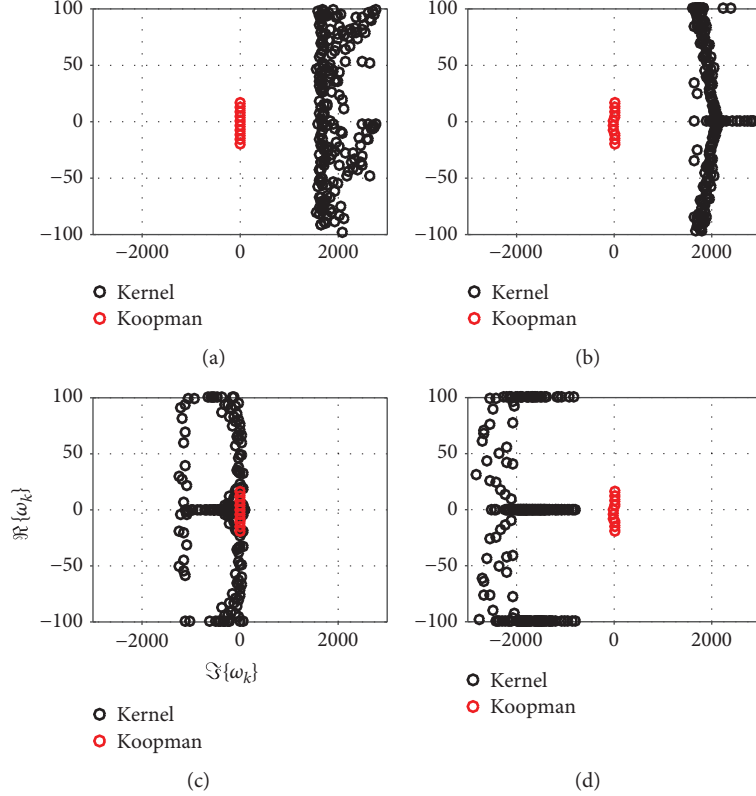


FIGURE 5: (a)–(d) Koopman spectra of the four kernels considered in (37) and (38a)–(38c), respectively. The red spectra are the Koopman spectra generated from the rank $r = 10$ observable $g_1(x)$ which provides an exceptionally accurate reconstruction of the NLS dynamics.

Specifically, the choice of kernel must be carefully selected for either the extended or kernel DMD to give anything reasonable. Cross-validation techniques could potentially be used to select a suitable kernel for applications of interest. It could also ensure that overfitting of the data does not occur. In either case, this simple example should serve as a strong cautionary tale for using kernel techniques in Koopman theory unless results are carefully cross validated. In contrast, our judiciously selected variables produce reasonable and parsimonious results which can be easily cross validated, saving a great deal of computational effort in producing improved performance over DMD.

5.3. Cubic Quintic Ginzburg-Landau Equation. A second and more difficult example for the DMD and Koopman theory to characterize is the cubic-quintic Ginzburg-Landau equation (CQGLE). The CQGLE is a canonical model from mathematical physics that exhibits a wide range of nonlinear spatio-temporal dynamics, including spatio-temporal periodicity and chaos. The evolution equation for CQGLE is given by [45]

$$iu_t + \left(\frac{1}{2} - i\tau\right)u_{xx} - iku_{xxx} + (1 - i\beta)|u|^2u + (\nu - i\sigma)|u|^4u - i\gamma u = 0, \quad (39)$$

where the state variable $u(x, t)$ is a function of space and time. Unlike the NLS equation, the CQGLE is not Hamiltonian and integrable, rather there are significant effects from

dissipation and gain effects, both linear and nonlinear. An example solution generated from the CQGLE is illustrated in Figure 6. This breather-type solution, although simple looking, does not have a simple low-rank representation like the NLS two-soliton breather. Indeed, the singular value decay suggests that a large number of modes are required for an accurate reconstruction. Importantly, the temporal evolution of the POD modes, which can be extracted from the columns of the \mathbf{V} matrix of the SVD, shows that the temporal evolution of the dynamics is quite complicated. This makes it difficult for the DMD approximation since it relies on approximating the temporal evolution by simple Fourier modes in time.

We again consider two additional observables

$$\mathbf{g}_3(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ |\mathbf{x}|^2\mathbf{x} \\ |\mathbf{x}|^4\mathbf{x} \end{bmatrix}, \quad (40a)$$

$$\mathbf{g}_4(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ |\mathbf{x}|^4 \end{bmatrix}. \quad (40b)$$

The first observable $g_3(x)$ is motivated by the form of the nonlinearity in the CQGLE equation. The second, $g_4(x)$, is chosen to have a quartic nonlinearity. The latter of the observables has no special relationship to the governing equations. And as before, a wide range of other

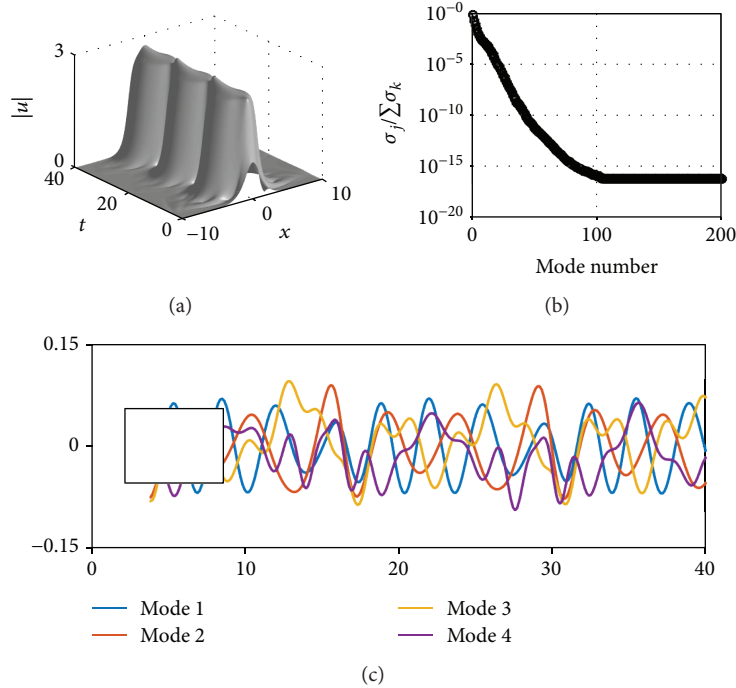


FIGURE 6: Spatio-temporal breather solution of the CQGLE equation for the parameter regime $\tau = 0.08$, $\kappa = 0$, $\beta = 0.66$, $\nu = -0.1$, $\sigma = -0.1$, and $\gamma = -0.1$. Although the dynamics illustrated (a) look relatively simple, the singular value decay (b) shows that a large number of modes are required to reconstruct the fine spatio-temporal features of the nonlinear evolution. Moreover, the temporal dynamics of the first four POD modes, which are extracted from the columns of the \mathbf{V} matrix of the SVD (c), characterize a complicated temporal behavior for the individual modes. Unlike the NLS example, a low-rank approximation does not work well for reconstructing the dynamics. The simulation was performed over $t \in [0, 40]$ with 301 equally spaced snapshots taken. The domain was discretized with $n = 512$ points on a domain $x \in [-10, 10]$.

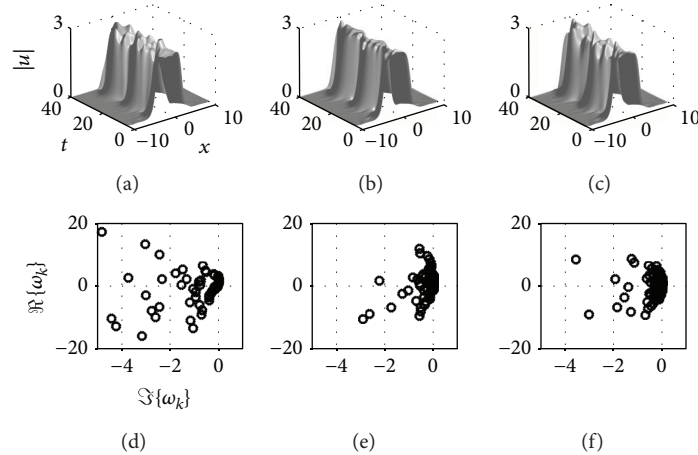


FIGURE 7: Reconstruction of the CQGLE dynamics of Figure 6(a) using a rank $r = 60$ expansion. Shown are (a) a standard DMD approximation $g_{\text{DMD}}(x)$, (b) the CQGLE motivated $g_3(x)$, and (c) a quartic observable $g_4(x)$. The Koopman spectra for each observable are demonstrated in (d), (e), and (f) which accompany the observables of (a), (b), and (c), respectively. Note that the observable $g_3(x)$ produces spectra which are most approximately purely imaginary which is expected of the periodic evolution. A visual inspection shows that the observable $g_3(x)$ produces the best reconstruction.

randomly selected nonlinear observables produce similar results to $g_4(x)$.

The DMD and Koopman reconstructions of the dynamics of the CQGLE are illustrated in Figure 7. As with the NLS example, the CQGLE motivated $g_3(x)$ gives the best

reconstruction. Importantly, the spectra generated are closest to the imaginary axis, which is expected for the periodic spatio-temporal dynamics observed. Indeed, the DMD algorithm, or its Koopman variant applied to observables, ideally generates a purely imaginary spectrum.

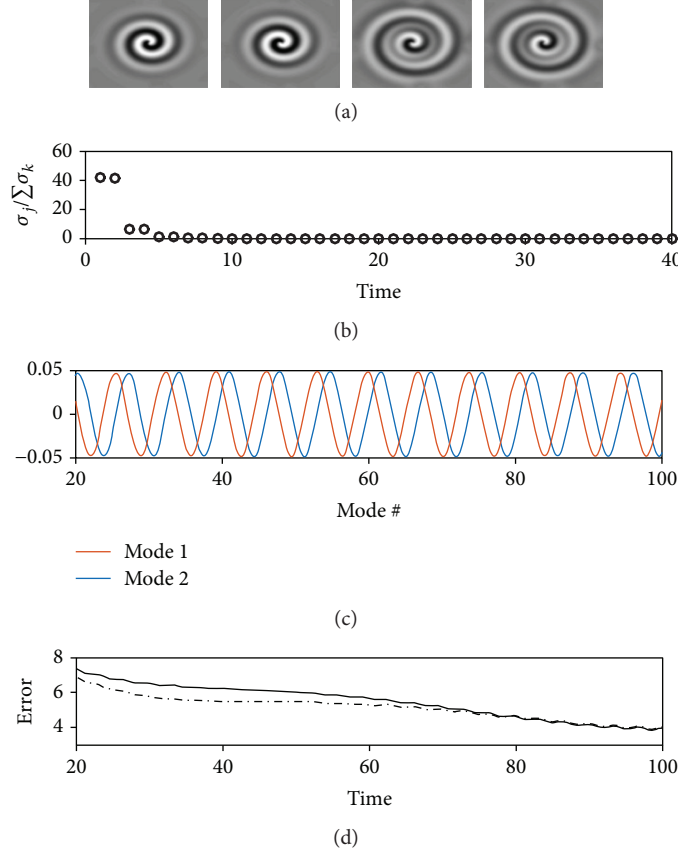


FIGURE 8: Diagnostic features of the reaction diffusion system (41a) and (41b) with $\beta = 1$ and $D = 0.1$. (a) The first four POD modes from sampling snapshots of the spiral wave dynamics. For this data, the snapshots were collected once the spiral wave was formed. Additionally, a spatial filter of the form $f(x, y) = \exp[-0.01(x^2 + y^2)]$ was applied in order to remove the effects of the periodic boundary conditions on the dynamics. (b) There is a two-mode dominance in the dynamics. (c) The temporal dynamics of the first two POD modes, which are extracted from the columns of the \mathbf{V} matrix of the SVD. For the spiral waves, these temporal modes are nearly perfect sinusoids. (d) A comparison of the error between the DMD reconstruction (solid line) and reaction-diffusion-inspired observable $g_5(u, v)$ (dash-dot line). Note that there is very little difference in the reconstruction error in contrast to the NLS and CQGLE examples. The simulation was performed over $t \in [0, 100]$ with 1001 equally spaced snapshots taken. The domain was discretized with $n = 128$ points on the domains $x \in [-20, 20]$ and $y \in [-20, 20]$.

5.4. *Reaction Diffusion System.* As a final example, we consider the $\lambda - \omega$ reaction-diffusion system [46]

$$u_t = D\nabla^2 u + \lambda(A)u - \omega(A)v, \quad (41a)$$

$$v_t = D\nabla^2 v + \omega(A)u + \lambda(A)v, \quad (41b)$$

where $A^2 = u^2 + v^2$, $\omega(A) = -\beta A^2$, $\lambda(A) = 1 - A^2$, $D = 0.1$, $\beta = 1$, and periodic boundaries are applied. This model generates spiral wave solutions which are sustained in the reaction-diffusion process. For processing the data, a spatial filter of the form $f(x, y) = \exp[-0.01(x^2 + y^2)]$ is applied to the snapshots. This removes the boundary effects from computing on a square domain.

Figure 8 shows key characteristics of the evolution dynamics and the decomposition architecture. In Figure 8(a), the first four POD modes are shown. These are the dominant modes of the dynamics associated with the spiral wave, containing approximately 99% of the total variance.

In the low-rank approximation applied, only the first two modes $r = 2$ are used. Importantly, the temporal evolution of the POD modes, which can be extracted from the columns of the \mathbf{V} matrix of the SVD, shows that the temporal evolution of the dynamics is almost purely sinusoidal. This makes it exceptionally easy for the DMD approximation since it relies on approximating the temporal evolution by simple Fourier modes in time. Indeed, given the simple sinusoidal evolution in time, the direct application of DMD is not improved upon by using additional observables. Specifically, we can consider the reaction-diffusion motivated observable

$$\mathbf{g}_5(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ (\mathbf{u}^2 + \mathbf{v}^2)\mathbf{u} \\ (\mathbf{u}^2 + \mathbf{v}^2)\mathbf{v} \end{bmatrix}, \quad (42)$$

where \mathbf{u} and \mathbf{v} are the discretized and reshaped vectors formed from numerically solving the reaction diffusion system. Figure 8(d) shows the reconstruction error for the DMD algorithm compared with the Koopman reconstruction using the observable (42). The error for both is comparable, which is in contrast to the previous examples of NLS and CQGLE where a significant increase in accuracy was achieved using well-selected variables. The fact is that given the almost perfectly sinusoidal low-rank nature of the temporal dynamics, the DMD algorithm simply does not require additional observables to produce an exceptional approximation.

6. Outlook on Koopman Theory for PDEs

Koopman analysis is a remarkable theoretical architecture with applicability to a wide range of nonlinear dynamical systems and PDEs. It combines a number of innovations across disciplines, including dimensionality-reduction techniques, manifold learning, linear operator theory, and dynamical systems. Although the abstract architecture provides a tremendously compelling viewpoint on how to transform nonlinear dynamical systems to infinite-dimensional linear dynamics, significant challenges remain in positing an appropriate set of observables for construction of the Koopman operator. If good candidate observables can be found, then the DMD algorithm can be enacted to compute a finite-dimensional approximation of the Koopman operator, including its eigenfunctions, eigenvalues, and Koopman modes. With a judicious choice of observables, these computed quantities can often lead to physically interpretable spatio-temporal features of the complex system under consideration.

We have demonstrated the application of Koopman theory on several canonical nonlinear PDEs: Burgers' equation, the nonlinear Schrödinger equation, the cubic-quintic Ginzburg-Landau equation, and a reaction-diffusion system. For Burgers' equation, the well-known Cole-Hopf transformation provides a critical link to an explicit calculation of the Koopman operator for a nonlinear PDE. Indeed, we show that the Koopman operator and associated observables can be trivially constructed from knowledge of the Cole-Hopf transformation. In contrast, choosing linear state observables for Burgers' yields a DMD approximation which is accurate but lacks the clear physical interpretation of the exact Koopman reduction. Although the NLS equation can similarly be linearized via the inverse scattering transform, the transform and its inverse are technically difficult to compute for arbitrary initial conditions. Instead, we demonstrate that the selection of an observable that is motivated by the nonlinearity of the governing PDE gives a remarkably accurate Koopman reduction. Indeed, the Koopman eigenfunctions and eigenvalues provide an approximation that is nearly equivalent to the accuracy of the numerical simulation itself. Importantly, for the NLS example, we also demonstrate that poor choices of observables are significantly worse than the DMD approximation. And for the case of observables chosen with a kernel method, the resulting spectra and eigenfunctions are highly inaccurate and nonrobust, suggesting that

such generic techniques as kernel methods may face challenges for use in observable selection. Importantly, the cross validation of the EDMD methods is critically important as the large number of variables used to describe the dynamics, most of which do not have any physical interpretability, must be carefully tuned. Like NLS, the CQGLE model can be similarly improved by variables motivated by the nonlinearity of the PDE. In contrast, the reaction-diffusion system shows that the standard DMD approximation is difficult to improve upon given that the temporal dynamics are almost purely sinusoidal. Such sinusoidal temporal evolution is ideal for the DMD approximation. Even if the governing PDE is not known, symmetries and/or conservation laws can help inform appropriate choices of a parsimonious set of observables. If nothing is known of the physics, then the standard EDMD remains a viable strategy for producing a model, even if interpretability and generalizability is typically lost. This remains an open direction of research which is beyond the current manuscript.

The results presented here provide a prescriptive algorithm for variable selection. Specifically, we recommend the following heuristic measures for a variable selection algorithm: (i) Upon performing the SVD of the data matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ of snapshots of the state space x_j , evaluate the temporal nature of the dominant modes via the columns of the matrix \mathbf{V} . If the dominant columns of \mathbf{V} are approximately sinusoidal, then the standard DMD algorithm should be used (see the reaction-diffusion example). (ii) If the dominant temporal behavior is not sinusoidal, then select observables motivated by the nonlinearity of the governing PDE (see NLS and CQGLE examples). (iii) If the governing PDE is unknown or the method in (ii) is performing poorly, then enact the cross-validated EDMD architecture. All three methods, DMD, judiciously chosen observables with DMD, and EDMD, are all important components of a robust strategy for evaluating nonlinear PDE dynamics.

Ultimately, the selected observables do not need to exactly linearize the system, but they should provide a method for transforming a strongly nonlinear dynamical system to a weakly nonlinear dynamical system. In practice, this is all that is necessary to make the method viable and informative. The results presented here are simultaneously compelling and concerning, highlighting the broader outlook of the Koopman method in general. Specifically, the success of the method will hinge on one issue: selection of observables. If principled techniques, from expert-in-the-loop knowledge, the form of the governing equation, or information about the manifold on which the data exists, can be leveraged to construct suitable observables, then Koopman theory should provide a transformative method for nonlinear dynamical systems and PDEs. We posit that sparse statistical regression techniques from machine learning may provide a path forward towards achieving this goal of selecting quality observables [23, 47]. Failing this, the Koopman architecture may have a limited impact in the mathematical sciences. Because of the importance of identifying meaningful observables, this is an exciting and growing area of research, especially given new developments in machine learning that may provide a robust and principled approach to observable

selection. For those interested in pursuing the EDMD architecture further, we recommend the recent text [12] which highlights many aspects of the current work and some of the structure of the EDMD algorithm that makes it computationally tractable. Included in the book is a link to all codes used in this manuscript.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

J. N. Kutz would like to acknowledge support from the Air Force Office of Scientific Research (FA9550-17-1-0329). S. L. Brunton and J. N. Kutz acknowledge support from the Defense Advanced Research Projects Agency (DARPA HR0011-16-C-0016). J.L. Proctor would like to thank Bill and Melinda Gates for their active support of the Institute for Disease Modeling and their sponsorship through the Global Good Fund.

References

- [1] A. J. Majda, "Challenges in climate science and contemporary applied mathematics," *Communications on Pure and Applied Mathematics*, vol. 65, no. 7, pp. 920–948, 2012.
- [2] S. L. Brunton and B. R. Noack, "Closed-loop turbulence control: progress and challenges," *Applied Mechanics Reviews*, vol. 67, article 50801, no. 5, 2015.
- [3] S. Ganguli and H. Sompolinsky, "Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis," *Annual Review of Neuroscience*, vol. 35, no. 1, pp. 485–508, 2012.
- [4] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.
- [5] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, p. 115, 2009.
- [6] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.
- [7] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, vol. 197, no. 1–2, pp. 101–133, 2004.
- [8] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, no. 1–3, pp. 309–325, 2005.
- [9] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, article 047510, 2012.
- [10] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annual Review of Fluid Mechanics*, vol. 45, no. 1, pp. 357–378, 2013.
- [11] J. Nathan Kutz, S. L. Brunton, D. M. Luchtenburg, C. W. Rowley, and J. H. Tu, "On dynamic mode decomposition: theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, 2014.
- [12] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, Society for Industrial and Applied Mathematics, 2016.
- [13] M. J. Ablowitz and H. Segur, *Solitons and the Inverse Scattering Transform*, Society for Industrial and Applied Mathematics, 1981.
- [14] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [15] I. G. Kevrekidis, C. W. Rowley, and M. O. Williams, "A kernel-based method for data-driven koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2015.
- [16] C. Schütte, P. Koltai, and S. Klus, "On the numerical approximation of the Perron-Frobenius and Koopman operator," *Journal of Computational Dynamics*, vol. 3, no. 1, pp. 1–12, 2016.
- [17] S. Klus, F. Nüske, P. Koltai et al., "Data-driven model reduction and transfer operator approximation," *Journal of Nonlinear Science*, vol. 28, no. 3, pp. 985–1010, 2018.
- [18] F. Noé and F. Nüske, "A variational approach to modeling slow processes in stochastic dynamical systems," *Multiscale Modeling & Simulation*, vol. 11, no. 2, pp. 635–655, 2013.
- [19] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, "Variational approach to molecular kinetics," *Journal of Chemical Theory and Computation*, vol. 10, no. 4, pp. 1739–1752, 2014.
- [20] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters*, vol. 72, no. 23, pp. 3634–3637, 1994.
- [21] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, "Identification of slow molecular order parameters for Markov model construction," *The Journal of Chemical Physics*, vol. 139, no. 1, article 15102, 2013.
- [22] B. O. Koopman and J. v. Neumann, "Dynamical systems of continuous spectra," *Proceedings of the National Academy of Sciences*, vol. 18, no. 3, pp. 255–263, 1932.
- [23] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS One*, vol. 11, no. 2, article e0150171, 2016.
- [24] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Society for Industrial and Applied Mathematics, 2005.
- [25] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, "Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems," *SIAM Journal on Applied Dynamical Systems*, vol. 13, no. 4, pp. 1716–1732, 2014.
- [26] S. Sargsyan, S. L. Brunton, and J. N. Kutz, "Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries," *Physical Review E*, vol. 92, no. 3, article 33304, 2015.

- [27] D. Amsallem, R. Tezaur, and C. Farhat, "Real-time solution of linear computational problems using databases of parametric reduced-order models with arbitrary underlying meshes," *Journal of Computational Physics*, vol. 326, pp. 373–397, 2016.
- [28] Y. Choi, D. Amsallem, and C. Farhat, "Gradient-based constrained optimization using a database of linear reduced-order models," 2015, <http://arxiv.org/abs/1506.07849>.
- [29] B. Peherstorfer and K. Willcox, "Online adaptive model reduction for nonlinear systems via low-rank updates," *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. A2123–A2150, 2015.
- [30] B. Peherstorfer and K. Willcox, "Dynamic data-driven reduced-order models," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 21–41, 2015.
- [31] B. Peherstorfer and K. Willcox, "Detecting and adapting to parameter changes for reduced models of dynamic data-driven application systems," *Procedia Computer Science*, vol. 51, pp. 2553–2562, 2015.
- [32] E. Kaiser, B. R. Noack, L. Cordier et al., "Cluster-based reduced-order modelling of a mixing layer," *Journal of Fluid Mechanics*, vol. 754, pp. 365–414, 2014.
- [33] B. Kramer, P. Grover, P. Boufounos, S. Nabi, and M. Benosman, "Sparse sensing and DMD-based identification of flow regimes and bifurcations in complex flows," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 2, pp. 1164–1196, 2017.
- [34] C. W. Rowley, M. O. Williams, and I. G. Kevrekidis, "Dynamic mode decomposition and the Koopman operator: algorithms and applications," *IPAM, UCLA*, 2014.
- [35] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [36] S. M. J. Ham, D. D. Lee, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Twenty-first international conference on Machine learning - ICML '04*, Banff, Alberta, Canada, July 2004.
- [37] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.
- [38] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, pp. 1244–1249, Washington, D.C., 2001.
- [39] W.-H. Steeb and F. Wilhelm, "Non-linear autonomous systems of differential equations and Carleman linearization procedure," *Journal of Mathematical Analysis and Applications*, vol. 77, no. 2, pp. 601–611, 1980.
- [40] K. Kowalski and W.-H. Steeb, *Nonlinear Dynamical Systems and Carleman Linearization*, World Scientific, 1991.
- [41] S. P. Banks, "Infinite-dimensional Carleman linearization, the Lie series and optimal control of non-linear partial differential equations," *International Journal of Systems Science*, vol. 23, no. 5, pp. 663–675, 1992.
- [42] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.
- [43] E. Hopf, "The partial differential equation $u_t + uu_x = \mu_{xx}$," *Communications on Pure and Applied Mathematics*, vol. 3, no. 3, pp. 201–230, 1950.
- [44] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, no. 3, pp. 225–236, 1951.
- [45] J. N. Kutz, "Mode-locked soliton lasers," *SIAM Review*, vol. 48, no. 4, pp. 629–678, 2006.
- [46] J. D. Murray, *Mathematical Biology. II Spatial Models and Biomedical Applications {Interdisciplinary Applied Mathematics V. 18}*, Springer-Verlag New York Incorporated, 2001.
- [47] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

Research Article

Long-Time Predictive Modeling of Nonlinear Dynamical Systems Using Neural Networks

Shaowu Pan  and Karthik Duraisamy 

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA

Correspondence should be addressed to Karthik Duraisamy; kdur@umich.edu

Received 31 May 2018; Revised 27 August 2018; Accepted 9 September 2018; Published 2 December 2018

Academic Editor: Francisco Chinesta

Copyright © 2018 Shaowu Pan and Karthik Duraisamy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study the use of feedforward neural networks (FNN) to develop models of nonlinear dynamical systems from data. Emphasis is placed on predictions at long times, with limited data availability. Inspired by global stability analysis, and the observation of strong correlation between the local error and the maximal singular value of the Jacobian of the ANN, we introduce Jacobian regularization in the loss function. This regularization suppresses the sensitivity of the prediction to the local error and is shown to improve accuracy and robustness. Comparison between the proposed approach and sparse polynomial regression is presented in numerical examples ranging from simple ODE systems to nonlinear PDE systems including vortex shedding behind a cylinder and instability-driven buoyant mixing flow. Furthermore, limitations of feedforward neural networks are highlighted, especially when the training data does not include a low dimensional attractor. Strategies of data augmentation are presented as remedies to address these issues to a certain extent.

1. Introduction

The need to model dynamical behavior from data is pervasive across science and engineering. Applications are found in diverse fields such as in control systems [1], time series modeling [2], and describing the evolution of coherent structures [3]. While data-driven modeling of dynamical systems can be broadly classified as a special case of system identification [4], it is important to note certain distinguishing qualities: the learning process may be performed off-line, physical systems may involve very high dimensions, and the goal may involve the prediction of long-time behavior from limited training data.

Artificial neural networks (ANN) have attracted considerable attention in recent years in domains such as image recognition in computer vision [5, 6] and in control applications [7]. The success of ANNs arises from their ability to effectively learn low-dimensional representations from complex data and in building relationships between features and outputs. Neural networks with a single hidden layer and nonlinear activation function are guaranteed to be able to predict

any Borel measurable function to any degree of accuracy on a compact domain [8].

The idea of leveraging neural networks to model dynamical systems has been explored since the 1990s. ANNs are prevalent in the system identification and time series modeling community [9–12], where the mapping between inputs and outputs is of prime interest. Billings et al. [13] explored connections between neural networks and the nonlinear autoregressive moving average model (NARMAX) with exogenous inputs. It was shown that neural networks with one hidden layer and sigmoid activation function represent an infinite series consisting of polynomials of the input and state units. Elanayar and Shin [14] proposed the approximation of nonlinear stochastic dynamical systems using radial basis feedforward neural networks. Early work using neural networks to forecast multivariate time series of commodity prices [15] demonstrated its ability to model stochastic systems without knowledge of the underlying governing equations. Tsung and Cottrell [16] proposed learning the dynamics in phase space using a feedforward neural network with time-delayed coordinates.

Paez and Urbina [17–19] modeled a nonlinear hardening oscillator using a neural network-based model combined with dimension reduction using canonical variate analysis (CVA). Smaoui [20–22] pioneered the use of neural networks to predict fluid dynamic systems such as the unstable manifold model for bursting behavior in the 2-D Navier-Stokes and the Kuramoto-Sivashinsky equations. The dimensionality of the original PDE system is reduced by considering a small number of proper orthogonal decomposition (POD) coefficients [23]. Interestingly, similar ideas of using principal component analysis for dimension reduction can be traced back to work in cognitive science by Elman [24]. Elman also showed that knowledge of the intrinsic dimensions of the system can be very helpful in determining the structure of the neural network. However, in the majority of the results [20–22], the neural network model is only evaluated a few time steps from the training set, which might not be a stringent performance test if longer time predictions are of interest.

ANNs have also been applied to chaotic nonlinear systems that are challenging from a data-driven modeling perspective, especially if long time predictions are desired. Instead of minimizing the pointwise prediction error, Bakker et al. [25] satisfied the Diks’ criterion in learning the chaotic attractor. Later, Lin et al. [26] demonstrated that even the simplest feedforward neural network for nonlinear chaotic hydrodynamics can show consistency in the time-averaged characteristics, power spectra, and Lyapunov exponent between the measurements and the model.

A major difficulty in modeling dynamical systems is the issue of memory. It is known that even for a Markovian system, the corresponding reduced-dimensional system could be non-Markovian [27, 28]. In general, there are two main ways of introducing memory effects in neural networks. First, a simple workaround for feedforward neural networks (FNN) is to introduce time delayed states in the inputs [11]. However, the drawback is that this could potentially lead to an unnecessarily large number of parameters [29]. To mitigate this, Bakker [25] considered following Broomhead and King [30] in reducing the dimension of the delay vector using weighted principal component analysis (PCA). The second approach uses output or hidden units as additional feedback. As an example, Elman’s network [29] is a recurrent neural network (RNN) that incorporates memory in a dynamic fashion.

Miyoshi et al. [31] demonstrated that recurrent RBF networks have the ability to reconstruct simple chaotic dynamics. Sato and Nagaya [32] showed that evolutionary algorithms can be used to train recurrent neural networks to capture the Lorenz system. Bailer-Jones et al. [33] used a standard RNN to predict the time derivative in discrete or continuous form for simple dynamical systems; this can be considered an RNN extension to Tsung’s phase space learning [16]. Wang et al. [34] proposed a framework combining POD for dimension reduction and long-short-term memory (LSTM) recurrent neural networks and applied it to a fluid dynamic system.

We limit ourselves to feedforward neural networks, since there are still many unanswered questions about modeling

dynamical systems even in this simplest form. It is known that time delayed FNNs closely resemble simple RNNs trained with teacher forcing [35]. Further, RNNs are not easy to train since standard training algorithms (e.g., back propagation through time [36]) are likely to introduce stronger overfitting than FNN due to vanishing gradients [35]. Recently, sparse regression (SINDy) [3, 4] has gained popularity as a tool for data-driven modeling. The idea is to search for a sparse representation of a linear combination of functions selected from a library. In this work, we will compare it with FNN-based models and highlight some differences.

The paper is organized as follows: the problem description is provided in Section 2 and the mathematical formulation of standard and Jacobian-regularized FNNs is presented in Section 3. Results and discussion are presented in Section 4. We first present a comparison with SINDy for simple dynamical systems. Then, we highlight the importance of stabilization to control the global error of predicted trajectory and the impact of Jacobian regularization. Finally, we apply the model in a nonlinear PDE system where a low dimensional attractor is not realized and discuss the limitations of black-box modeling of dynamical system and propose data augmentation as remedies. Conclusions are drawn in Section 5.

2. Problem Description

Consider a dynamical system in Euclidean space \mathbb{R}^M which is described by a continuously differentiable function $\phi : \mathbb{R} \times \mathbb{R}^M \mapsto \mathbb{R}^M$, where $\phi(t, \mathbf{x}) = \phi_t(\mathbf{x})$. The state $\mathbf{x} \in \mathbb{R}^M$ satisfies the composition relation $\phi_{t+s} = \phi_t \circ \phi_s$ for $t, s \in \mathbb{R}$ and $\phi_0 : \mathbb{R}^M \mapsto \mathbb{R}^M$ is the identity function. ϕ_t is the t map of the flow described by a vector function $\mathbf{F}_c : \mathbb{R}^M \mapsto \mathbb{R}^M$ as

$$\dot{\mathbf{x}} = \mathbf{F}_c(\mathbf{x}). \quad (1)$$

Similarly, one can define a discrete dynamical system induced by the above smooth dynamical system by considering a constant time step $\Delta t \in \mathbb{R}$ and a state transition map $F_d(\mathbf{x}) = \phi_{\Delta t}(\mathbf{x}) : \mathbb{R}^M \mapsto \mathbb{R}^M$ such that

$$\mathbf{x}^{n+1} = F_d(\mathbf{x}^n). \quad (2)$$

Equivalently, one can rewrite the above system as

$$\frac{(\mathbf{x}^{n+1} - \mathbf{x}^n)}{\Delta t} = \frac{(F_d(\mathbf{x}^n) - \mathbf{x}^n)}{\Delta t} \triangleq \mathbf{F}_r(\mathbf{x}^n), \quad (3)$$

where $\mathbf{F}_r : \mathbb{R}^M \mapsto \mathbb{R}^M$ resembles a first order solution [33] to \mathbf{F}_c .

Our goal is to find an approximation to the dynamics, either (i) in a discrete sense \mathbf{F}_d , given the data $\mathcal{D} = \{\mathbf{x}^i\}_{i=0}^{N-1}$ uniformly sampled from a trajectory given initial condition $\mathbf{x}^0 = \mathbf{x}(t=0)$ or (ii) in a continuous sense \mathbf{F}_c , given the data $\mathcal{D} = \{(\mathbf{x}^i, \dot{\mathbf{x}}^i)\}_{i=0}^{N-1}$, where N is the number of data points. It must be mentioned that—as highlighted in the result section—data does not have to be collected on the same trajectory.

Depending on the way one defines the training and testing set, two types of problems are considered in the current work.

- (1) Prediction of a certain trajectory starting from an initial condition that is different from the training trajectories
- (2) Prediction of the future trajectory given the past information of the trajectory as training data

Conservatively speaking, the success of tackling the first of the above problems requires the trajectories in the training data to be a representative of the distribution in the region of interest, which may or may not be feasible depending on how informative the data is. In the context of modeling dynamical systems, it is often implied in previous literature [22] that the initial condition of unseen testing data is not far away from the training data. The second problem can also be difficult since it will challenge the effectiveness of the model as past information might not be sufficient for the model to be predictive on unseen data. Again, it is often implied in the previous works [20, 34, 37] that successful predictions are often accompanied by an underlying low dimensional attractor so the past states as training data can be collected until it becomes a representative of the future.

3. Mathematical Modeling Framework

In this section, we first define performance metrics of the approximation to the dynamics f ; then, introduce the standard FNN model and the Jacobian-regularized FNN model. Finally, techniques to mitigate overfitting are described.

3.1. Definitions of Error Metrics. To measure the prediction error for each sample in an a priori sense (i.e., given exact \mathbf{x}^i), we define the local error vector $\xi_{\text{local}}^i \in \mathbb{R}^M$ for the i -th sample $(\mathbf{x}^i, \mathbf{y}^i)$ as

$$\xi_{\text{local}}^i = \mathbf{y}^i - f(\mathbf{x}^i), \quad (4)$$

where $\mathbf{y}^i \in \mathbb{R}^M$ is the i -th target to learn from the i -th feature $\mathbf{x}^i \in \mathbb{R}^M$. For example, the feature is the state vector at the i -th step, \mathbf{x}^i and the target can be \mathbf{x}^{i+1} for discrete dynamical system or $\dot{\mathbf{x}}^i$ for continuous dynamical system.

Then, we can define local error at the i -th sample by

$$e_{\text{local}}^i = \left\| \xi_{\text{local}}^i \right\|_2 = \left\| \mathbf{y}^i - f(\mathbf{x}^i) \right\|_2, \quad (5)$$

where $\|\cdot\|_2 : \mathbb{R}^M \mapsto [0, +\infty)$ is the vector 2-norm, i.e., l^2 norm, and $|\cdot| : \mathbb{R} \mapsto [0, \infty)$ is the absolute value.

We can further define the local error of the i -th sample for the j -th component as shown by

$$e_{\text{local},j}^i = \left| y_j^i - f(\mathbf{x}^i)_j \right|. \quad (6)$$

The local error assumes that the i -th input feature \mathbf{x}_i is predicted accurately. On the other hand, the global error

vector is defined by equation (7), in which $\hat{\mathbf{x}}^i$ is obtained by *iterative prediction*, i.e., a posteriori *evaluation*, at the i -th step from an initial condition through either time integration or transition function as a discrete map. That is, $\hat{\mathbf{x}}^i$ is obtained from $f(\hat{\mathbf{x}}^{i-1})$ in a recursive sense as follows:

$$\xi_{\text{global}}^i = \mathbf{x}^i - \hat{\mathbf{x}}^i. \quad (7)$$

Similarly, the global error is defined by

$$e_{\text{global}}^i = \left\| \xi_{\text{global}}^i \right\|_2 = \left\| \mathbf{x}^i - \hat{\mathbf{x}}^i \right\|_2, \quad (8)$$

and for the j -th component specifically by

$$e_{\text{global},j}^i = \left| x_j^i - \hat{x}_j^i \right|. \quad (9)$$

Further, to obtain a holistic view of the model performance in feature space, if F_d or F_c is known, either in the continuous or discrete case, we can define stepwise error as

$$e_{\text{stepwise}}(\mathbf{x}) = \left\| F_{c,d}(\mathbf{x}) - f(\mathbf{x}) \right\|_2. \quad (10)$$

Note that e_{stepwise} is not restricted by the training or testing trajectory, but it can be evaluated arbitrarily in the region of interest.

Finally, we consider the uniform averaged coefficient of determination R^2 as a scalar metric for measuring regression performance

$$R^2 = \frac{1}{M} \sum_{j=0}^{M-1} R_j^2, \quad (11)$$

where R_j^2 is given by

$$R_j^2 = R^2(y_j, \tilde{y}_j) = 1 - \frac{\sum_{i=0}^{n_{\text{sample}}-1} (y_j^i - \tilde{y}_j^i)^2}{\sum_{i=0}^{n_{\text{sample}}-1} (y_j^i - \bar{y}_j)^2}, \quad (12)$$

where n_{sample} is the number of samples in the validation data, $\bar{y}_j = 1/n_{\text{sample}} \sum_{i=0}^{n_{\text{sample}}-1} y_j^i$ and $\tilde{y}^i = f(\mathbf{x}^i)$ is the prediction of f based on the i -th feature \mathbf{x}^i .

3.2. Feedforward Neural Network Model

3.2.1. Basic Model: Densely Connected Feedforward Neural Network. The basic model approximates F_c in equation (1) for the continuous case and F_r in equation (3) in the discrete case using a feedforward neural network. The existence of an arbitrarily accurate feedforward neural network approximation to any Borel measurable function given the enough number of hidden units is guaranteed from the property of the universal approximator [38]. It should be noted that our basic model is related to Tsung's phase-space-learning model [16]. If the Markovian assumption is adopted, the

training feature matrix snapshots X and training target matrix snapshots Y are as follows:

$$X = \begin{bmatrix} x_1^0 & x_2^0 & x_3^0 & \dots & x_M^0 \\ x_1^1 & x_2^1 & x_3^1 & \dots & x_M^1 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{N-1} & x_2^{N-1} & x_3^{N-1} & \dots & x_M^{N-1} \end{bmatrix} \in \mathbb{R}^{N \times M}, \quad (13)$$

and

$$Y = \begin{bmatrix} y_1^0 & y_2^0 & y_3^0 & \dots & y_M^0 \\ y_1^1 & y_2^1 & y_3^1 & \dots & y_M^1 \\ \dots & \dots & \dots & \dots & \dots \\ y_1^{N-1} & y_2^{N-1} & y_3^{N-1} & \dots & y_M^{N-1} \end{bmatrix} \in \mathbb{R}^{N \times M}, \quad (14)$$

where M is the dimension of the state, N is the total number of snapshots of training data, learning target Y is the time derivative, and the subscript stands for the index of the component. Note that each component of the feature and target are normalized to zero mean and unit variance for better training performance in the neural network.

By generally constructing a densely connected feedforward neural network $f(\cdot): \mathbb{R}^M \mapsto \mathbb{R}^M$ with $L-1$ hidden layers and output layer as linear, the following recursive expression is defined for each hidden layer:

$$\eta^l = \sigma_l(W_l \eta^{l-1} + b_l), \quad l = 1, \dots, L-1, \quad (15)$$

where η^0 stands for the input of the neural network \mathbf{x} , $\eta^l \in \mathbb{R}^{n_l}$, $n_l \in \mathbb{N}^+$ is the number of hidden units in layer l , and σ_l is the activation function of layer l . Note that the output layer is linear, i.e., $\sigma_L(x) = x$:

$$f(\mathbf{x}; \mathbf{W}^L, \mathbf{b}^L) = \eta^L = W_L \eta^{L-1} + b_L, \quad (16)$$

where the parameters of the neural network are $\mathbf{W}^L = \{W_i\}_{i=1, \dots, L}$, $\mathbf{b}^L = \{b_i\}_{i=1, \dots, L}$.

For example, if we consider using two hidden layers where $L=3$ and the number of hidden units are the same, the full expression for the neural network model is given by

$$\hat{\mathbf{y}} = f(\mathbf{x}; \mathbf{W}, \mathbf{b}) = W_3 \sigma(W_2 \sigma(W_1 \mathbf{x} + b_1) + b_2) + b_3, \quad (17)$$

where $\mathbf{x} \in \mathbb{R}^M$ is the state of the dynamical system, i.e., the input to the neural network and $\hat{\mathbf{y}} \in \mathbb{R}^M$ is the modeling target, i.e., the output of neural network. $\sigma(\cdot): \mathbb{R} \mapsto \mathbb{R}$ is a nonlinear activation function. $W_1 \in \mathbb{R}^{n_h \times M}$, $W_2 \in \mathbb{R}^{n_h \times n_h}$, $W_3 \in \mathbb{R}^{M \times n_h}$. $b_1 \in \mathbb{R}^{n_h}$, $b_2 \in \mathbb{R}^{n_h}$, $b_3 \in \mathbb{R}^M$. Sets of weights and biases are $\mathbf{W}^3 = \{W_1, W_2, W_3\}$ and $\mathbf{b}^3 = \{b_1, b_2, b_3\}$. The problem is to find the set of parameters of \mathbf{W}^3 and \mathbf{b}^3 that result in the best approximation of the underlying ground truth (\mathbf{F}_e , \mathbf{F}_d , or \mathbf{F}_r). Under the framework of statistical learning, it is standard to perform empirical risk minimization (ERM) with mean-square-error loss. The set

up and parameters corresponding to the desired solution $f(\mathbf{x}, \mathbf{W}^*, \mathbf{b}^*)$ can be written as

$$\mathbf{W}^*, \mathbf{b}^* = \arg \min_{\mathbf{W}^3, \mathbf{b}^3} \frac{1}{|I_{\text{train}}|} \sum_{i \in I_{\text{train}}} \|f(\mathbf{x}^i; \mathbf{W}^3, \mathbf{b}^3) - \mathbf{y}^i\|_2^2, \quad (18)$$

where I_{train} is the index set of training data, and \mathbf{x}^i and \mathbf{y}^i correspond to the i -th feature target pair.

To deal with the high dimensionality of the optimization problem, we employ Adam [39], a gradient-based algorithm, which is essentially a mixture between momentum acceleration and rescaling parameters. The weights are initialized using a truncated normal distribution to potentially avoid saturation and use the automatic differentiation (AD) provided by Tensorflow [40] to compute the gradients. The neural network model is implemented in Python using the Tensorflow library [40]. Due to the nonconvex nature of equation (18), for such a high degree of freedom of parameters, one can only afford to find a local minimum. In practice, however, a good local minimum is usually satisfactory [35]. Hyperparameters considered in the current work for the basic model are the number of units for each hidden layer n_h and activation function $\sigma(\cdot)$.

Model selection for neural networks is an active research area [41–43]. Well-known methods involve grid search/random search [41]/Tree of Parzen Estimators (TPE) [42]/Bayesian optimization [43] with cross validation. We pursue the following trial-error strategy:

- (1) Given the number of training points, computing the number of equations to satisfy if the network overfits all the training data
- (2) Pick a neural network with uniform hidden layer structure to overfit the training data with the number of parameters in the network no more than 10% to 50% of the number in step 1
- (3) Keep reducing the size of neural network by either decreasing the hidden units or number of layers until the training and validation error are roughly the same order
- (4) For the choice of other hyperparameters, we simply perform grid search

3.2.2. Jacobian Regularized Model. In standard FNNs, minimizing mean-squared-error on the training data only guarantees model performance in terms of the local training error. It does not guarantee the reconstruction of even training trajectory in the *a posteriori* sense.

Here, we take a closer look at the error propagation in a dynamical system for the FNN model when evaluated in an iterative fashion, i.e., *a posteriori* sense. Without any loss of generality, considering the discrete case, after we obtain the model f , we can predict $\hat{\mathbf{x}}^{i+1}$ given $\hat{\mathbf{x}}^i$

$$\hat{\mathbf{x}}^{i+1} = f(\hat{\mathbf{x}}^i). \quad (19)$$

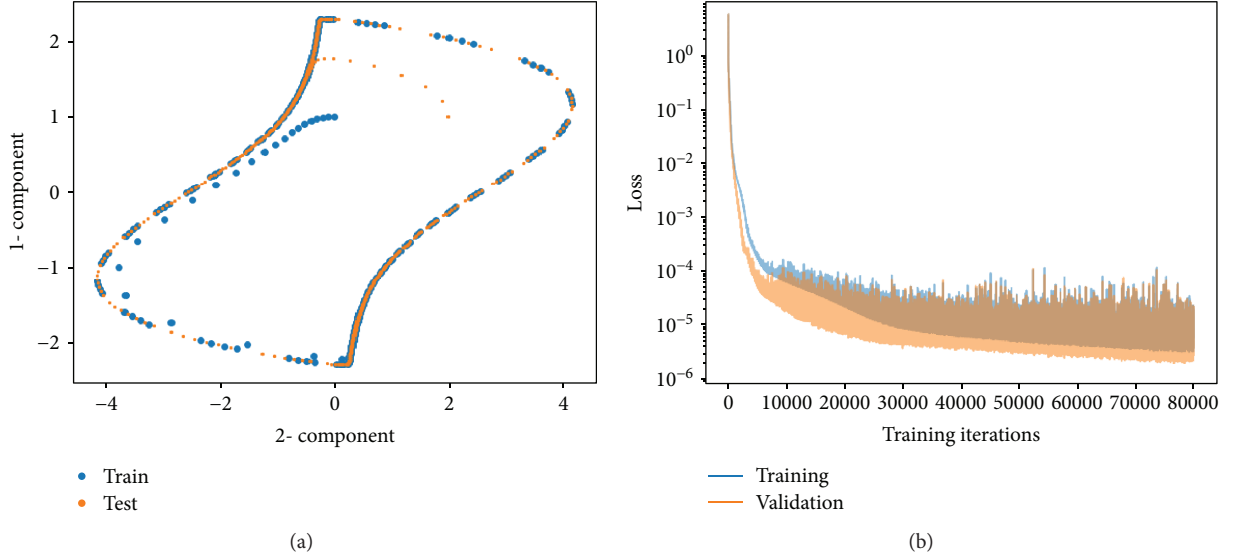


FIGURE 1: VDP case. (a) Distribution of training/testing sets. (b) Learning rate.

Moreover, given \mathbf{F}_d , we can find the $\xi_{\text{global}}^{i+1}$ given $\hat{\mathbf{x}}^i$ and ξ_{global}^i as follows

$$\xi_{\text{global}}^{i+1} = \mathbf{x}^{i+1} - \hat{\mathbf{x}}^{i+1} = \mathbf{F}_d(\mathbf{x}^i) - f(\hat{\mathbf{x}}^i) = \mathbf{F}_d(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) - f(\hat{\mathbf{x}}^i), \quad (20)$$

$$\begin{aligned} \xi_{\text{global}}^{i+1} &= \mathbf{F}_d(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) - f(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) \\ &\quad + f(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) - f(\hat{\mathbf{x}}^i). \end{aligned} \quad (21)$$

Consider a Taylor expansion of $f(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i)$ about $\hat{\mathbf{x}}^i$, we have

$$\begin{aligned} \xi_{\text{global}}^{i+1} &= \mathbf{F}_d(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) - f(\hat{\mathbf{x}}^i + \xi_{\text{global}}^i) + \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^i} \xi_{\text{global}}^i \\ &\quad + \frac{1}{2} \xi_{\text{global}}^i T H \xi_{\text{global}}^i + \dots, \end{aligned} \quad (22)$$

where H is the Hessian matrix evaluated at some point between $\hat{\mathbf{x}}^i$ and $\hat{\mathbf{x}}^i + \xi_{\text{global}}^i$.

Assuming $\|\xi_{\text{global}}^i\|_2 \ll 1$, $\|H\|_2$ is bounded, and the high order terms are negligible compared to the Jacobian term, we have

$$\begin{aligned} e_{\text{global}}^{i+1} &\leq e_{\text{local}}^i + \left\| \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^i} \left\| \xi_{\text{global}}^i \right\|_2 + o(e_{\text{global}}^i) \\ &\leq e_{\text{local}}^i + \left\| \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^i} \left\| \xi_{\text{global}}^i \right\|_F + o(e_{\text{global}}^i). \end{aligned} \quad (23)$$

TABLE 1: Hyperparameter configuration of the basic model: VDP case.

Layer structure	Activation function	Loss function	Optimizer	Learning rate
2-8-8-2	Swish	MSE	Adam	0.002

Similarly, in the continuous case, we have

$$\begin{aligned} e_{\text{global}}^{i+1} &\leq e_{\text{global}}^i + \int_{i\Delta t}^{(i+1)\Delta t} e_{\text{local}}^{\tau/\Delta t} d\tau + \int_{i\Delta t}^{(i+1)\Delta t} \left\| \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^i} \left\| e_{\text{global}}^{\tau/\Delta t} \right\| d\tau \\ &\quad + o(e_{\text{global}}^i \Delta t), \end{aligned} \quad (24)$$

$$\begin{aligned} e_{\text{global}}^{i+1} &\leq \left(1 + \Delta t \left\| \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}^i} \right) e_{\text{global}}^i + \int_{i\Delta t}^{(i+1)\Delta t} e_{\text{local}}^{\tau/\Delta t} d\tau \\ &\quad + o(e_{\text{global}}^i \Delta t). \end{aligned} \quad (25)$$

The right hand sides of equations (23) and (25) contain contributions from the global error and accumulation of local error. Optimization as in equation (18) can minimize the latter term, but not necessarily the former. This suggests that manipulating the eigenspectrum of the Jacobian might be beneficial for stabilization by suppressing the growth of the error. Due to the simplicity of computing the Frobenius norm compared to the 2-norm, we consider penalizing the Frobenius norm of the Jacobian of the neural network model. In the context of improving generalization performance of input-output neural network models, similar regularization has been also proposed by Rifai et al. [44]. It should be noted that our purpose is to achieve better error dynamics in a temporal sense,

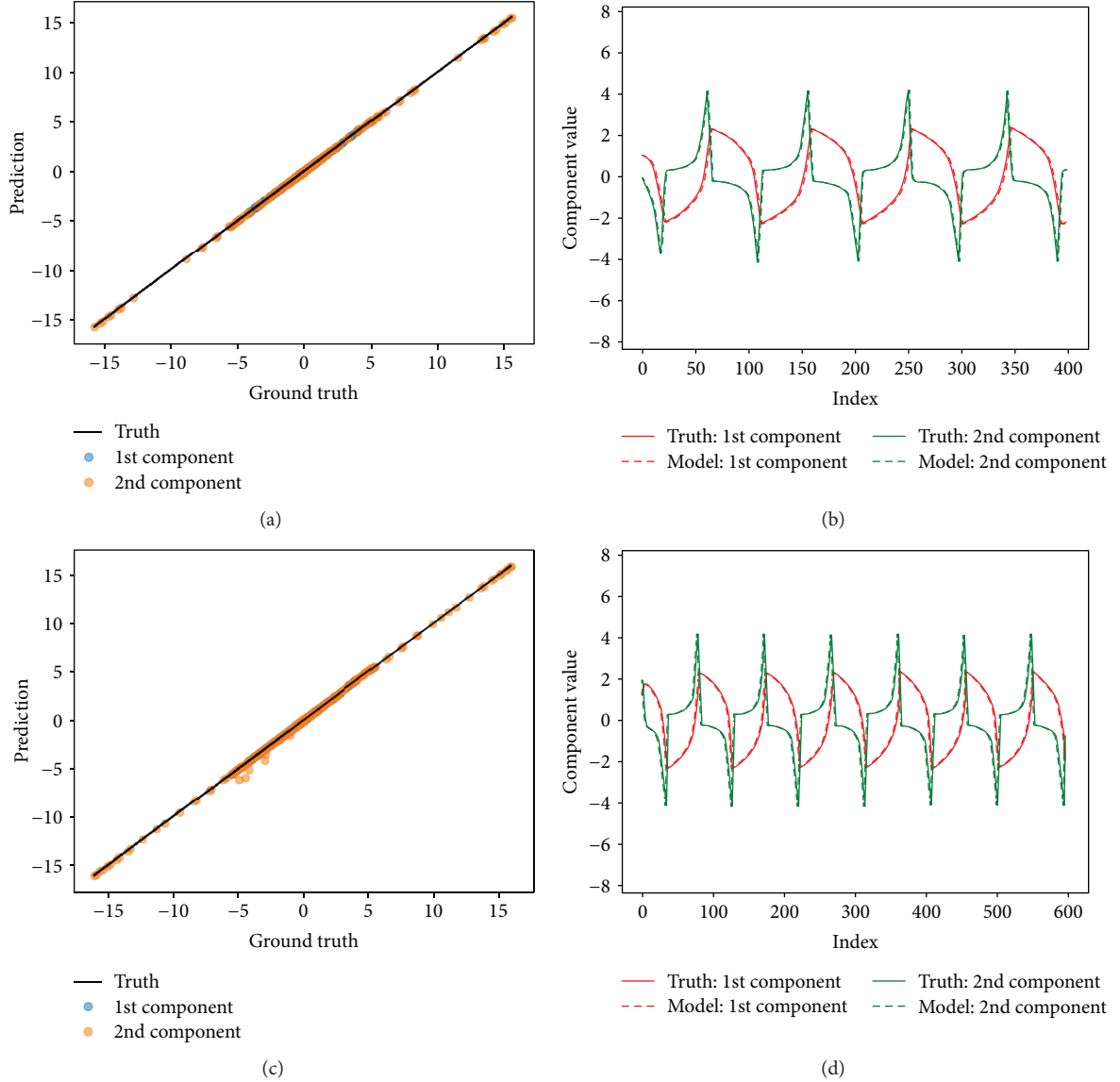


FIGURE 2: A priori and a posteriori result of the basic model on VDP case. (a, b) Training data. (c, d) Testing data. (a, c) A priori. (b, d) A posteriori.

which differs from the generalization goal in deep learning. Thus, one may seek a locally optimal solution that can suppress the growth in global error while minimizing the local error.

The regularized loss function inspired from the above discussion is thus

$$\mathbf{W}^*, \mathbf{b}^* = \arg \min_{\mathbf{W}, \mathbf{b}} \frac{1}{N_{\text{train}}} \sum_{i \in I_{\text{train}}} \|f(\mathbf{x}^i; \mathbf{W}, \mathbf{b}) - \mathbf{y}^i\|_2^2 + \lambda \|J(\mathbf{x}^i; \mathbf{W}, \mathbf{b})\|_F^2, \quad (26)$$

where J is the Jacobian of the neural network output with respect to the input and λ is a hyperparameter. On one hand, it should be noted that regularizing the Frobenius norm of the eigenspectrum of the Jacobian indirectly suppresses the magnitude of the eigenvalue of the Jacobian.

On the other hand, excessive weighting on the magnitude of the eigenvalue would lead to less weighting on local error, which might result in an undesirably large local error. Thus, λ should be set as a relatively small value without strongly impacting the model performance in an a priori sense.

3.3. Reducing Overfitting. Overfitting is a common issue in the training of machine learning models, and it arises when models tend to memorize the training data instead of generalizing true functional relations. In neural networks, overfitting can occur from poor local minima and is partially due to the unavoidable nonconvexity of an artificial neural network. Overfitting cannot be completely eliminated for most problems, given the NP-hard nature of the problem. Generally, overfitting can be controlled by three kinds of regularization techniques. The first follows the Occam's razor

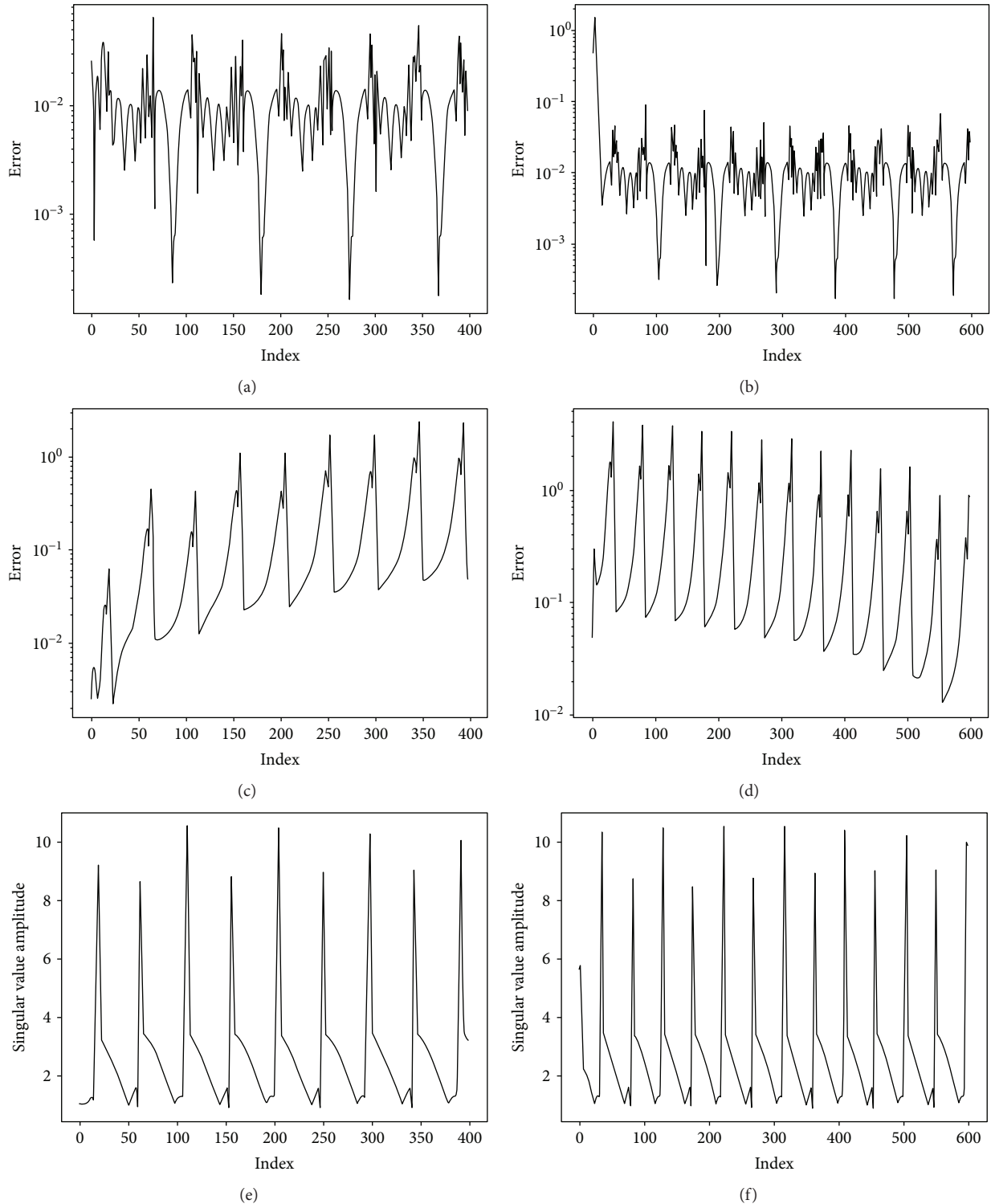


FIGURE 3: Variation of local error and global error: basic model on VDP case. (a, b) Local error. (c, d) Global error. (e, f) Maximal singular value of the Jacobian evaluated in a priori. (a, c, e) Training data. (b, d, f) Testing data.

principle, e.g., L1 sparsity regularization [3]. However, there is no guarantee that Occam's razor is appropriate for all cases, and finding the optimal sparsity level is often iterative. The second is to smooth the function, e.g., using weight decay [35]. The third type is especially suitable in iterative learning,

e.g., early stopping, which is a widely used strategy in the deep learning community [35]. In this work, we found validation-based early stopping to be sufficient. We split the data further into pure training and validation sets, and then monitor overfitting by measuring R^2 .

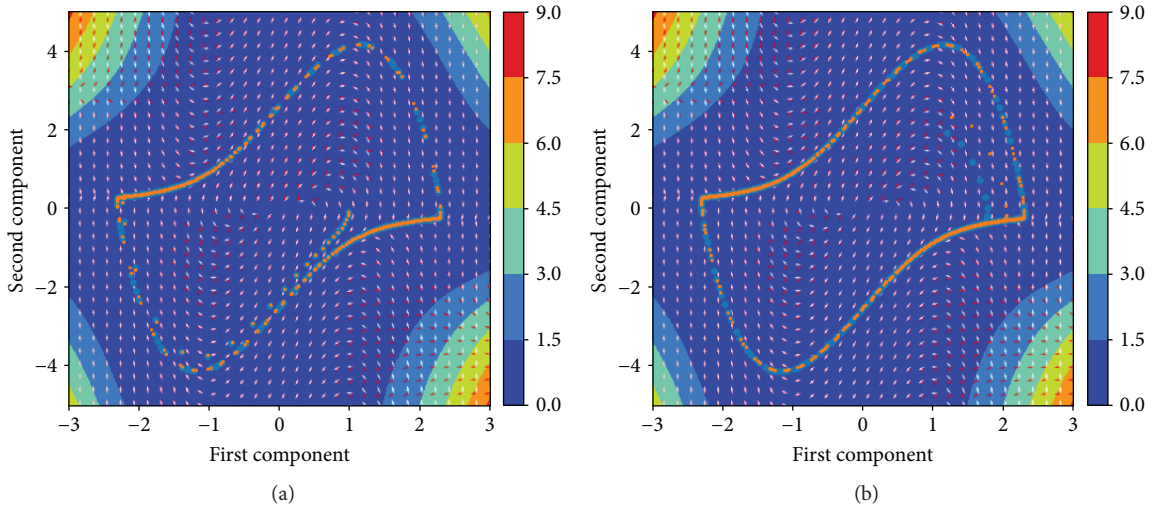


FIGURE 4: Stepwise error contour of the basic model on VDP case. (a) Training data. (b) Testing data. Blue dot: ground truth. Orange dot: prediction. White arrow: direction of target vector of ground truth. Red arrow: direction of target vector of prediction.

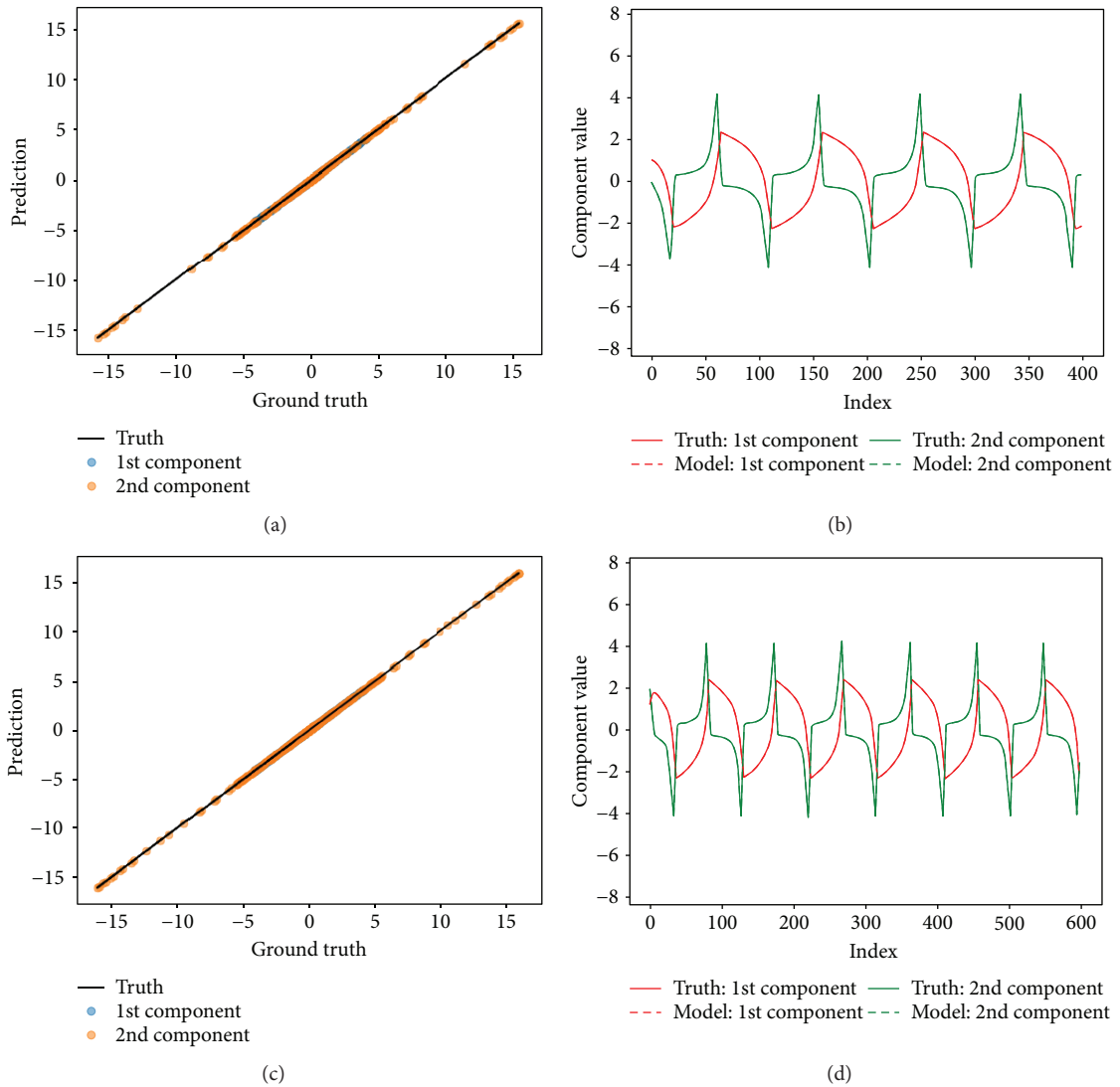


FIGURE 5: A priori and a posteriori result of SINDy on VDP case. (a, b) Training data. (c, d) Testing data. (a, c) A priori. (b, d) A posteriori.

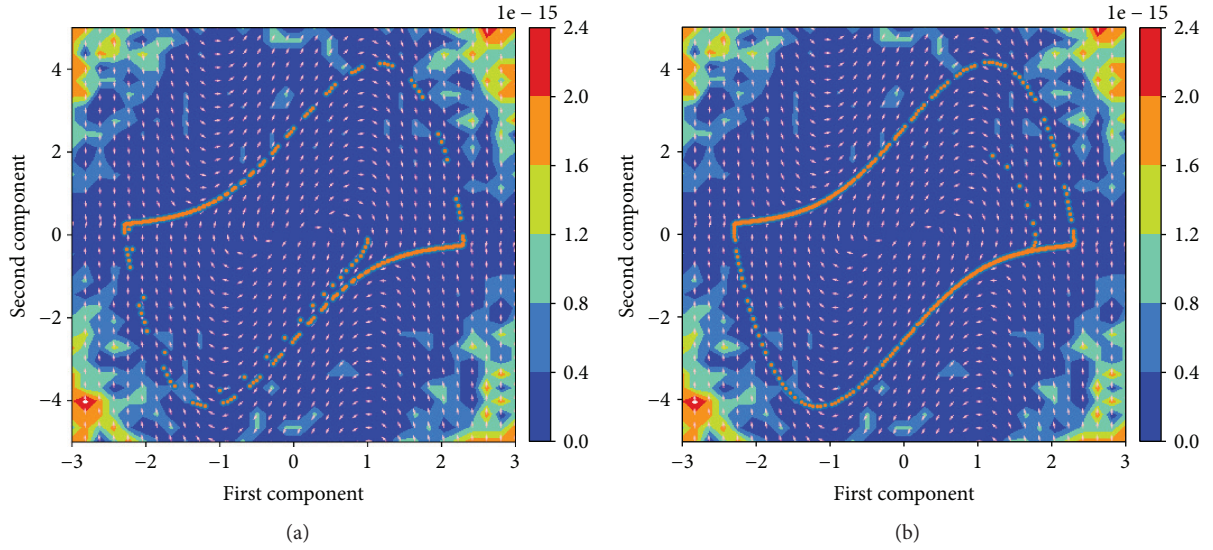


FIGURE 6: Stepwise error contour of SINDy on VDP case. (a) Training data. (b) Testing data. Blue dot: ground truth. Orange dot: prediction. White arrow: direction of target vector of ground truth. Red arrow: direction of target vector of prediction.

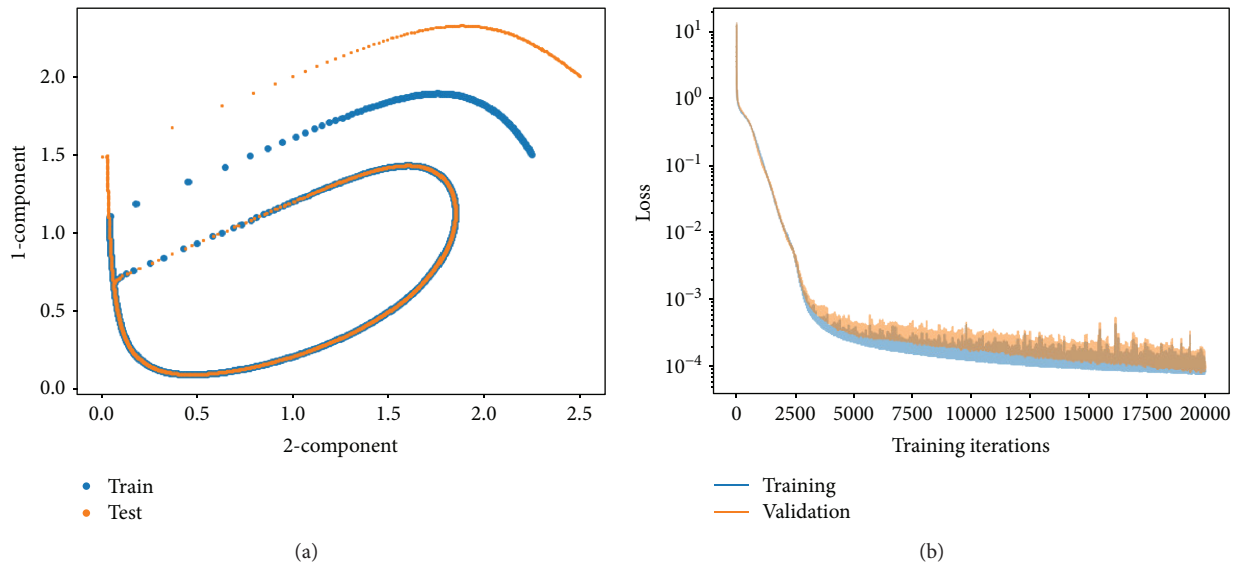


FIGURE 7: Nonrational nonpolynomial case. (a) Data distribution of training/testing feature. (b) Learning curve.

4. Results and Discussion

Given sequential training data, the capability of the basic FNN is first evaluated in two-dimensional dynamical systems with polynomial nonlinearities in Section 4.1 and nonpolynomial nonrational dynamics in Section 4.2. The basic model is compared with SINDy [3], which is a method that directly aims at learning functional models using L_1 sparse regression on a dictionary of candidate basis functions. In Section 4.3, we demonstrate that the basic model performs better than SINDy on the problem of incompressible flow behind a cylinder, in spite of the explicit addition of quadratic terms to the dictionary. In

TABLE 2: Hyperparameter configuration of the basic model: nonrational nonpolynomial case.

Layer structure	Activation function	Loss function	Optimizer	Learning rate
2-8-8-2	elu	MSE	Adam	0.005

addition, the local error is found to be strongly correlated with the maximal singular value of the Jacobian, thus serving as an inspiration for Jacobian regularization. In Section 4.4, we demonstrate the stabilizing aspect of

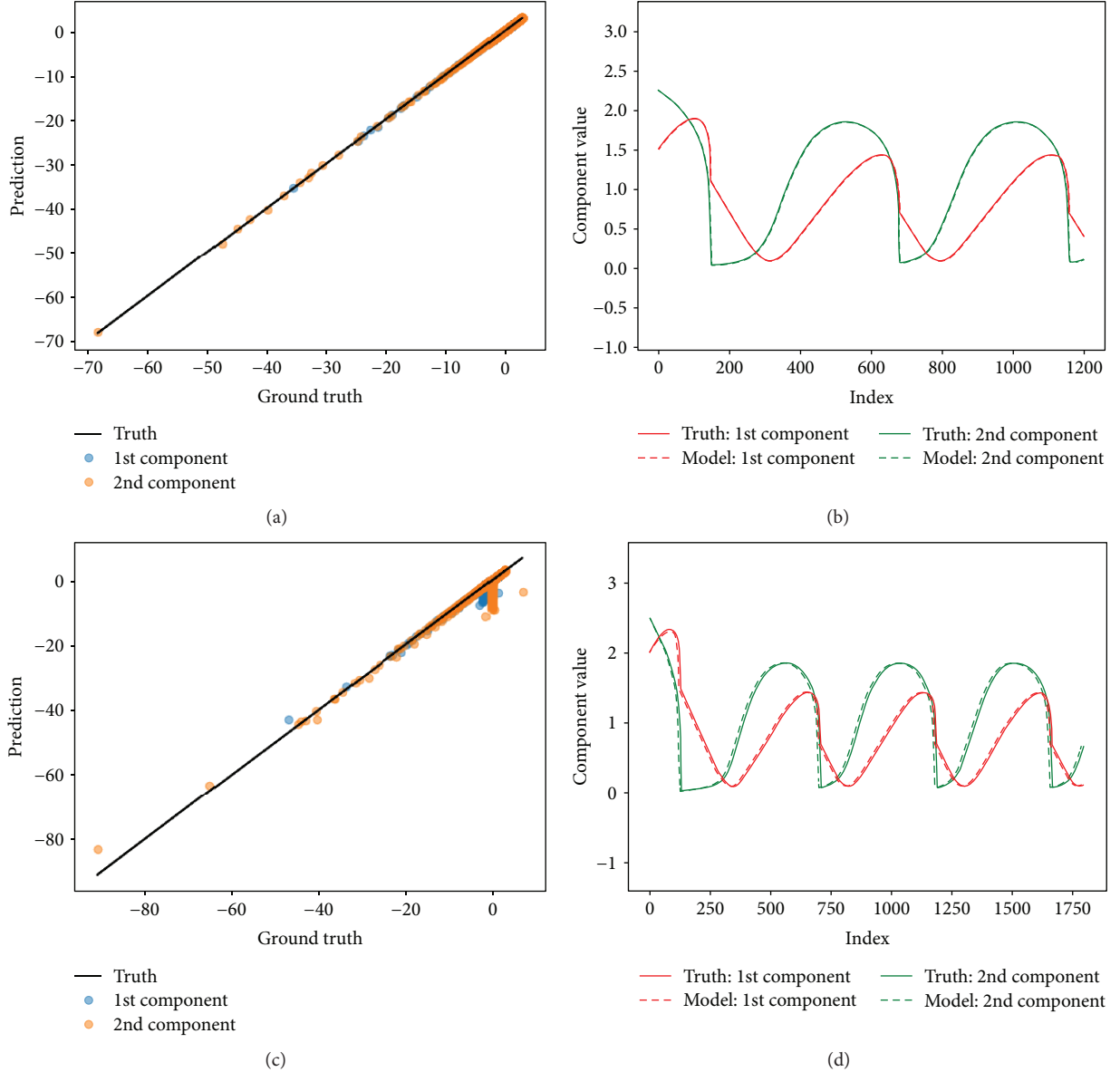


FIGURE 8: A priori and a posteriori result of the basic model on nonrational nonpolynomial case. (a, b) Training data. (c, d) Testing data. (a, c) A priori. (b, d) A posteriori.

Jacobian regularization for the problem of laminar wake behind a cylinder, where the system exhibits a low dimensional attractor. In Section 4.5, we assess the ability of our regularized FNN model to approximate a dynamically evolving high-dimensional buoyancy-driven mixing flow system that is a characteristic of flow physics driven by instabilities. The results show that, for systems that do not exhibit a low dimensional attractor, it is difficult for a black-box model to have satisfactory long-time prediction capabilities. In Section 4.6, we show that predictive properties can be improved by data augmentation in the state space of interest.

4.1. 2D Polynomial System: Van der Pol Oscillator. The first order forward discretized scheme of the Van der Pol (VDP) system is given by

$$\begin{pmatrix} x_1^{n+1} \\ x_2^{n+1} \end{pmatrix} = \begin{pmatrix} x_1^n \\ x_2^n \end{pmatrix} + \Delta t \begin{pmatrix} x_2^n \\ (\mu(1-x_1^n)x_2^n - x_1^n) \end{pmatrix}, \quad (27)$$

where $\Delta t = 0.1$ and $\mu = 2.0$. The modeling target is

$$\mathbf{y}^n = \begin{pmatrix} \frac{(x_1^{n+1} - x_1^n)}{\Delta t} \\ \frac{(x_2^{n+1} - x_2^n)}{\Delta t} \end{pmatrix} = \begin{pmatrix} x_2^n \\ (\mu(1-x_1^n)x_2^n - x_1^n) \end{pmatrix} = F_r(\mathbf{x}^n). \quad (28)$$

Our goal is to reproduce the dynamics governed by F_r from on data collected from a single trajectory. The loss

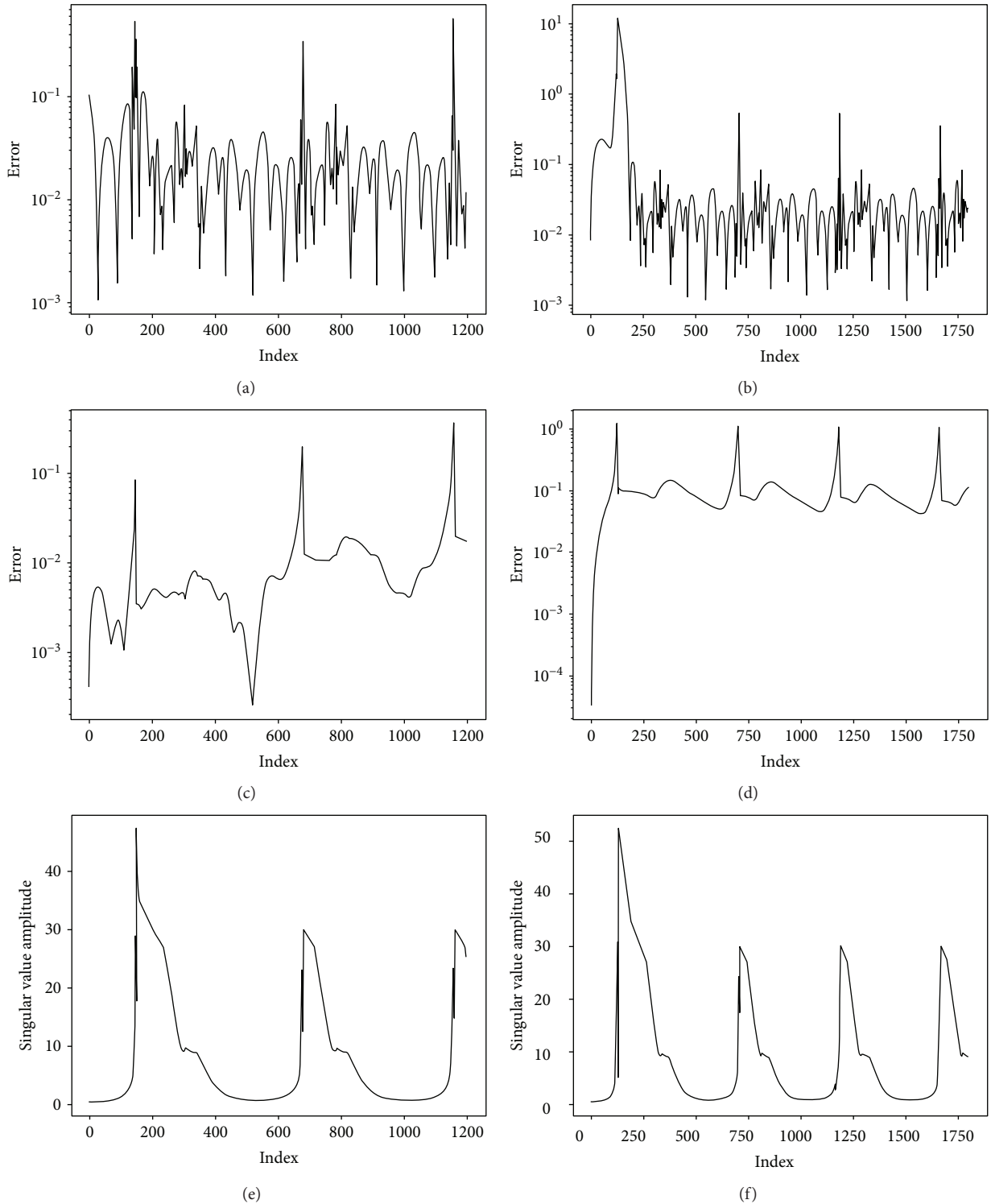


FIGURE 9: Variation of local error and global error: basic model on nonrational, nonpolynomial case. (a, b) Local error. (c, d) Global error. (e, f) Maximal singular value of the Jacobian evaluated in a priori. (a, c, e) Training data. (b, d, f) Testing data.

function of the basic model in equation (18) is optimized using training data from a single trajectory, containing 399 data points. Test data containing 599 points is generated using a different initial condition. The data distribution of the training and testing features is shown in Figure 1. Note that initially, a few test points (orange)

are away from the training data (blue) which require the model to perform extrapolation. Configuration of hyper-parameters is shown in Table 1. Data is normalized to zero mean and unit standard deviation for each component. We use minibatch training with batch size=64 and 80,000 epochs. The basic model consists of two hidden

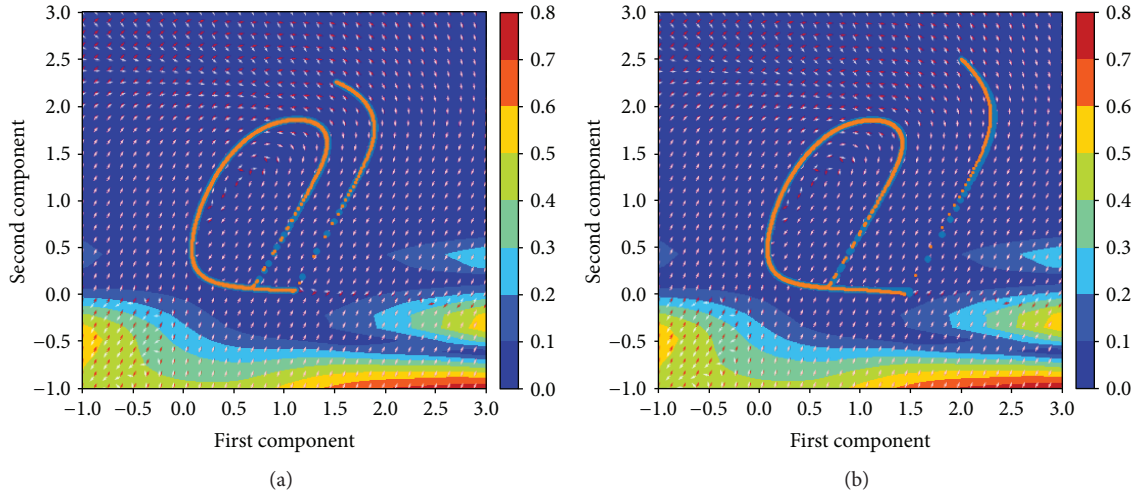


FIGURE 10: Stepwise error contour of the basic model on nonrational nonpolynomial case. (a) training data. (b) testing data. Blue dot: ground truth. Orange dot: prediction. White arrow: direction of target vector of ground truth. Red arrow: direction of target vector of prediction.

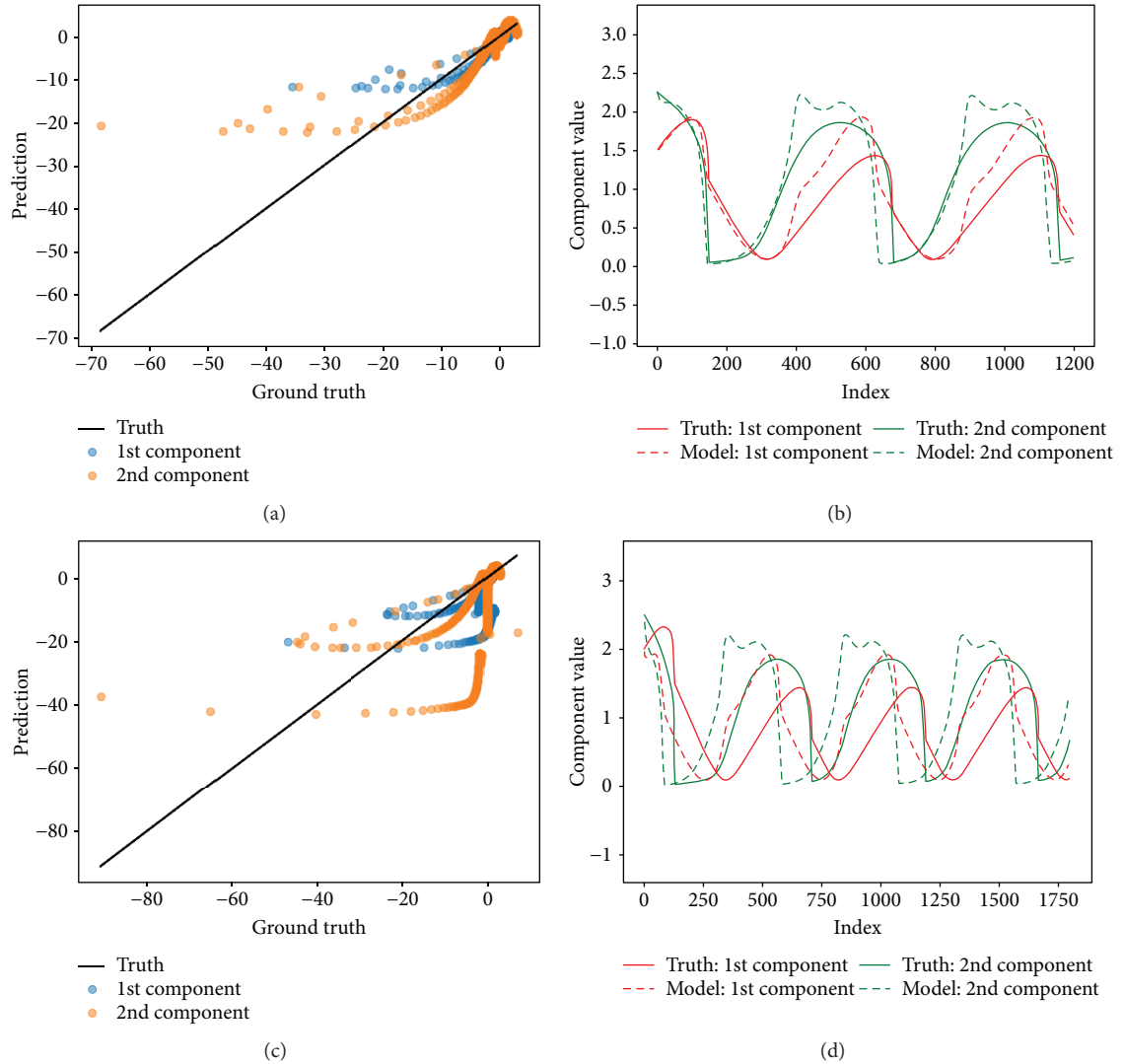


FIGURE 11: A priori and a posteriori result of SINDy on nonrational nonpolynomial case. (a, b) Training data. (c, d) Testing data. (a, c) A priori. (b, d) A posteriori.

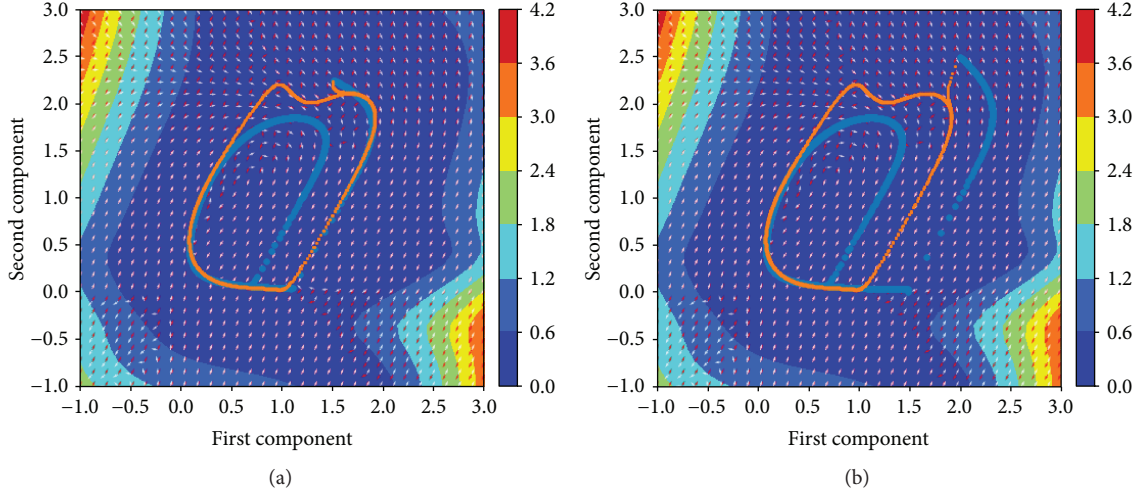


FIGURE 12: Stepwise error contour of SINDy on nonrational nonpolynomial case. (a) Training data. (b) Testing data. Blue dot: ground truth. Orange dot: prediction. White arrow: direction of target vector of ground truth. Red arrow: direction of target vector of prediction.

layers with each layer containing 8 hidden units. Two hidden layers are accompanied by Swish nonlinear activation as $\sigma(x) = x \cdot \text{sigmoid}(\beta x)$, where in practice, β is fixed as unity [45]. The output layer is linear. Randomly 20% of training data is used as a validation set, and we monitor the performance on the validation set as a warning of overfitting. In Figure 1, the learning curve suggests that the model is well-trained and overfitting is not observed.

Results of a priori and a posteriori prediction are shown in Figure 2. The basic model predicts the \mathbf{F}_r at each training point very well a priori, but slight phase lag is observed a posteriori in testing, which originates from the extrapolation of the testing data initially.

The variation of the local and global error together with the maximal singular value of the Jacobian is shown in Figure 3. For training data, e_{local} is observed to be relatively uniform as expected, since the objective optimized is MSE uniformly across all training data points. For testing data, e_{local} exhibits peak values near the beginning of the trajectory as expected, since the first few points are far away from the training data shown in Figure 1. Moreover, it is interesting to observe that in Figure 3, the peak of the temporal history of local/global error shows a strong correlation with the maximal singular value of the Jacobian.

Stepwise error contours are displayed in Figure 4. The region of large error close to red (implying the difference of the stepwise vector between neural network prediction and ground truth is large) is located near the corner of figure, where there is a dearth of training points. The model performs well near the training points as expected. In this case, since testing data is not very far away from the training data, good performance of extrapolation can be expected. However, we would like to note that there is a moderate amount of error associated with the vector direction in Figure 4 not only at the corners but also near the origin. This implies that a feedforward neural network can generalize to some extent, but with no guarantees, even in regions

enclosed by training data. The results also confirm that the known result for a dynamical system with an attractor, the neural network can reproduce the dynamics near the attractor [13, 25, 33, 37, 46].

With the prior knowledge that the system is polynomial in nature, one can use polynomial basis functions to extract the ground truth. To illustrate this, results obtained from SINDy [3] are shown in Figure 5, with threshold parameter as 2×10^{-4} , maximal polynomial order as 3, and no validation data set considered. As displayed in Figure 6, the excellent result of SINDy shows the advantage of finding the global features where the parameters obtained are not restricted to the scope of training data since the ground truth is governed by sparse polynomials.

4.2. 2D Nonpolynomial System: A Nonrational Nonpolynomial Oscillator. The success of SINDy is a consequence of the fact that the underlying system can be represented as a sparse vector in a predefined basis library such as that consisting of polynomial or rational functions [4]. Here, we choose a different case: a nonrational nonpolynomial oscillator with $\Delta t = 0.004$

$$\begin{pmatrix} x_1^{n+1} \\ x_2^{n+1} \end{pmatrix} = \begin{pmatrix} x_1^n \\ x_2^n \end{pmatrix} + \Delta t \left(2.5 - 100 \frac{x_1^n x_2^n}{1 + (x_2^n/0.52)^4} - 200 \frac{x_1^n x_2^n}{1 + (x_2^n/0.52)^4} + 9.2 - 2.3x_2^n - 1.28|x_2^n|^{3/2} \right). \quad (29)$$

Here, the basic model in equation (18) is optimized using 1199 data points of a single trajectory. Testing data contains 1799 points. Randomly, 20% of the training data is taken as the validation set, but also included in later evaluation. The feature distribution in phase space is shown in Figure 7. Hyperparameters are listed in Table 2 and 128

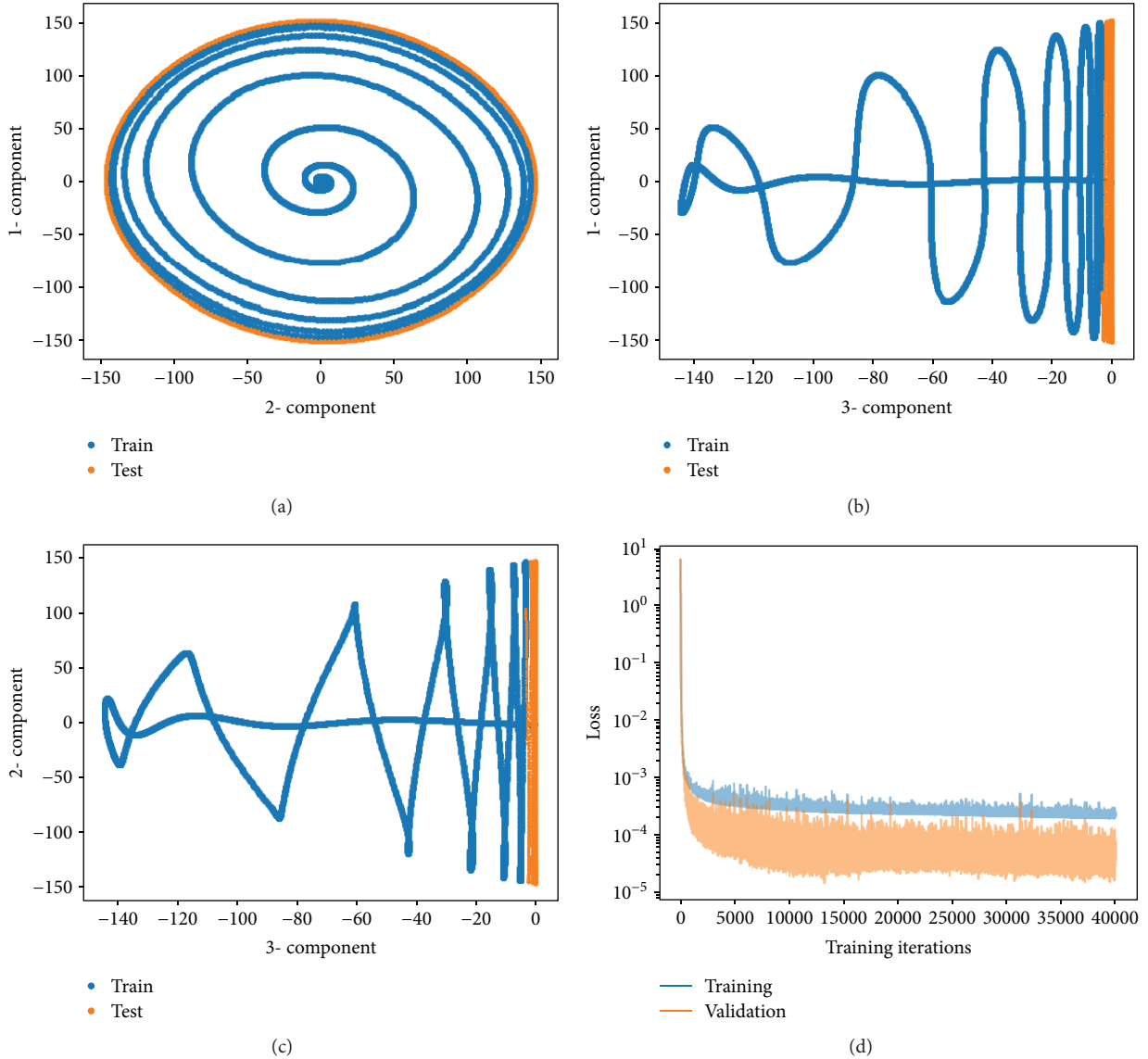


FIGURE 13: Flow in a cylinder wake. (a) Data distribution of x_1 vs. x_2 . (b) Data distribution of x_1 and x_3 . (c) Data distribution of x_2 and x_3 . (d) Learning curve.

minibatches and 20,000 epochs are used. The training error and validation error is also shown in Figure 7.

Results for a priori and a posteriori performance on training and testing data are shown in Figure 8. The training trajectory is perfectly reconstructed while the predictions show slight deviation.

The distribution of the local and global error is shown in Figure 9. Again, we observe that maximal local/global error correlates with the peaks of the maximal singular value of the Jacobian. It is interesting to note that the highest local testing error occurs at the peak of the maximal singular value of the Jacobian, instead of at the points close to the initial condition.

The error contour in Figure 10 shows that the stepwise error around the training trajectory is below 0.1. It is important to note that model performance deteriorates at places far

TABLE 3: Hyperparameter configuration of the basic model: flow in a cylinder wake.

Layer Structure	Activation Function	Loss Function	Optimizer	Learning rate
2-20-20-2	elu	MSE	Adam	0.001

away from the training trajectory, especially at the right corner shown in Figure 10.

For SINDy, the polynomial order is set to three and threshold as 2×10^{-4} . A priori and a posteriori validation for training and testing is shown in Figure 11. Correspondingly, the stepwise error contour displayed in Figure 12 shows the misfit for the region of interests ranging from -1 to 3 for both two components. Because there is no sparsity

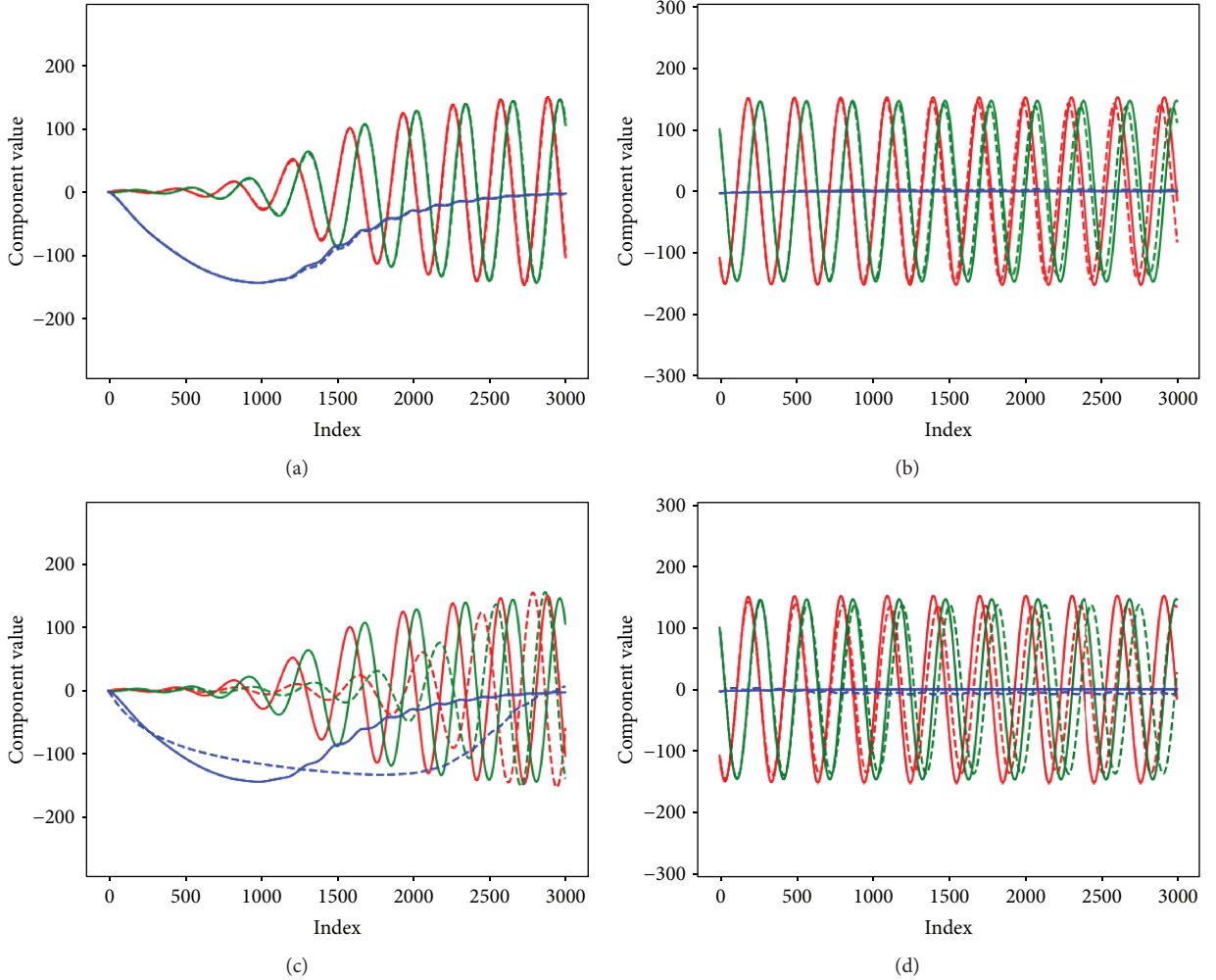


FIGURE 14: A posteriori comparison for flow in a cylinder wake between the basic model and SINDy. Red: first POD coefficient. Green: second POD coefficient. Blue: shift-mode coefficient. (a, b) Basic model. (c, d) SINDy. (a, c) Training data. (b, d) Testing data.

in polynomial basis in this case, it is expected that SINDy cannot reconstruct the dynamics correctly and would perform worse than the basic model of FNN. The implication is that for strongly nonpolynomial systems, neural networks are far more flexible compared to SINDy.

4.3. Nonlinear PDE System: Flow behind a Cylinder. In this section, we compare the basic model with SINDy in reconstructing the flow in a cylinder wake. The data is from Brunton et al. [3] which comes from an immersed boundary method solution [47] of the 2D incompressible N-S equations with $Re=100$ based on the cylinder diameter. The computational domain consists of a nonuniform grid with near-wall refinement. The inlet condition is uniform flow and the outlet is a convective boundary condition to allow the vorticity to exit the domain freely. Testing data is generated as a temporal extension of states that lie on a limit cycle at the boundary of training data, which indicates that this is not an extrapolation task. To work with such a high-dimensional nonlinear PDE system, we use the coefficients of two POD modes [23] and one “shift mode”, which

represents the shift of short-term averaged flow away from the POD space of the first two harmonic modes to reduce the spatial dimension. More details on POD and “shift-modes” are provided in references [23, 48]. Training and testing data is the same as in Brunton et al. [3] where the first 2999 snapshots in time are used for training, and a later 2994 snapshots used for testing. A random 10% of training snapshots is considered as validation set but also included in later evaluation. The distribution of training data and testing data is shown in Figure 13.

Hyperparameters of the basic model are shown in Table 3 with 40,000 epochs. For SINDy, the hyperparameters are the same as in the previous work [3]. As shown in Figure 14, for training data, SINDy reconstructs a smaller growth rate of oscillating behavior, while the basic model accurately reconstructs both the shift mode and two POD modes. For testing data, SINDy contains an observable phase lag for the time period concerned, while the basic model achieves an almost perfect match. This implies that the model obtained from SINDy, although much easier to interpret than neural network, is not the best model for this dynamical system in

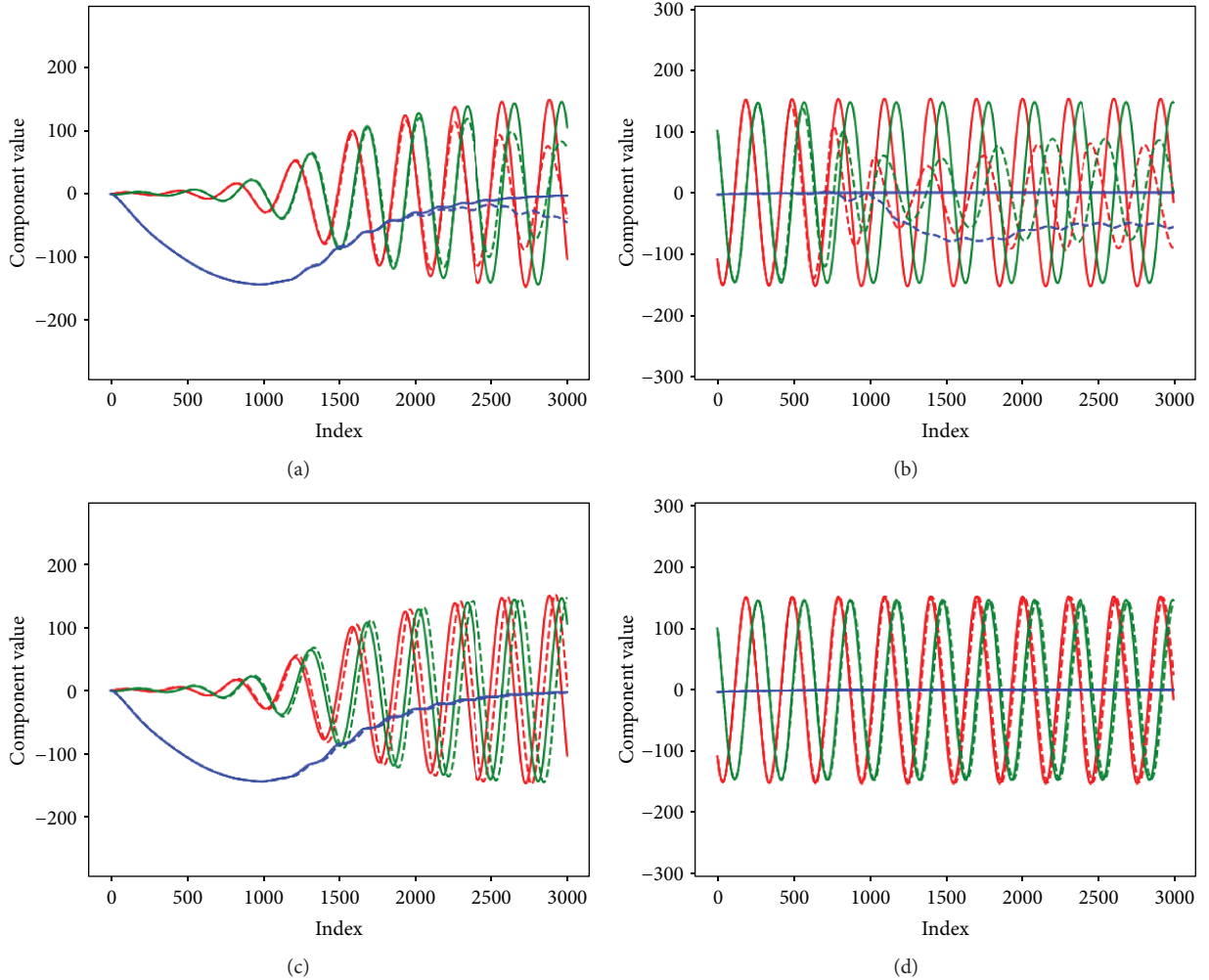


FIGURE 15: A posteriori comparison between basic and regularized models for cylinder wake. Red: first POD coefficient. Green: second POD coefficient. Blue: shift-mode coefficient. (a, b) Basic model. (c, d) Regularized model. (a, c) Training data. (b, d) Testing data.

terms of accuracy. However, we note that from the data distribution in Figure 13, the basic model performs as expected, as the training data covers the attractor well.

4.4. Stabilizing the Neural Network with Jacobian Regularization. Due to the nonconvexity of the optimization problem that arises in the solution of the basic model in equation (18), employing a stochastic gradient-descent type method might lead to a solution corresponding to a local minimum, which is often undesirable and difficult to avoid. Most works in the field of deep learning for feedforward neural networks focus on decreasing the impact of poor local minima to promote generalizability. However, in the context of modeling a dynamical system, as it is often assumed that the trajectory of interest is stable with respect to small disturbances [25], the model should be able to approximately reconstruct the training trajectory in the presence of local errors that arise at each step. This would require regularizing instabilities that could arise in a posteriori prediction. To have meaningful comparisons, random number seeds are fixed for initialization of weights and training data shuffling. Nevertheless, we observe that in some cases, for example, in

the previous case of the cylinder wake, an inappropriate choice of neural network configuration of the basic model, e.g., number of hidden units and type of activation function, can potentially lead to instability in a posteriori evaluation. Such instabilities may materialize even while reconstructing the training trajectory, while the corresponding a priori prediction is almost perfect. Previous work [16] explicitly ensured stability by simply adding more adjacent trajectories. Here, we take a different approach by adding a Jacobian regularization term in the cost function in equation (26).

In our numerical experiments, with a certain fixed random seed, it is observed that, when the layer structure is 2-20-20-2 with \tanh as activation function instead of elu , the basic model becomes numerically unstable after 2000 steps for training data which is displayed in Figure 15. Similar numerical instability is also observed in testing evaluations. However, for the same fixed random seed, the regularized model with $\lambda = 5 \times 10^{-5}$ shows numerically stable results with the same neural network configuration for both training and testing data.

The effectiveness of Jacobian regularization may be attributed to finding a balance between lowering the prediction

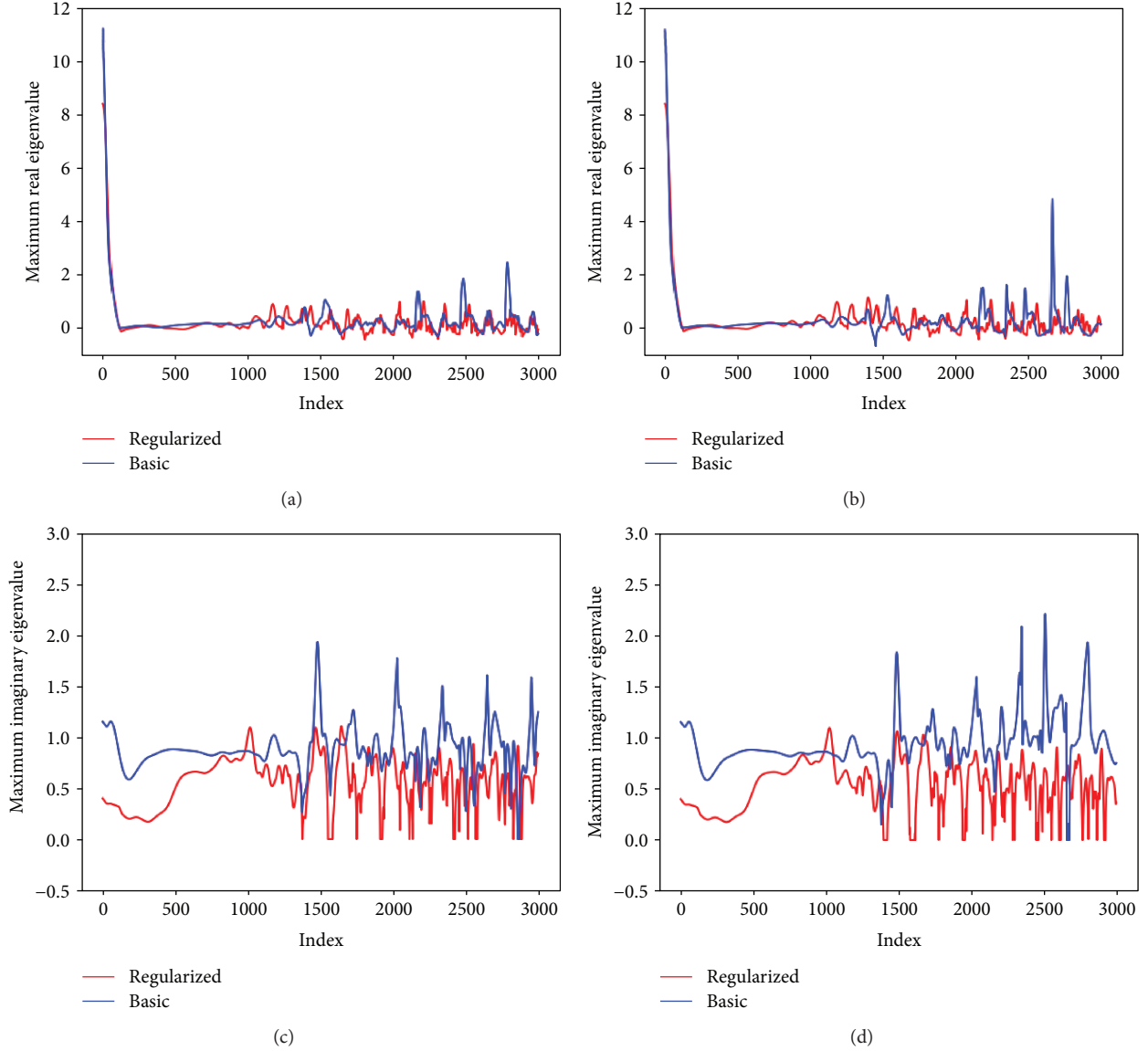


FIGURE 16: Flow in a cylinder wake. Comparison of eigenvalue of the Jacobian between regularized and basic model on training data. (a, b) Maximal real eigenvalue. (c, d) Maximal imaginary eigenvalue. (a, c) A priori. (b, d) A posteriori.

error, i.e., MSE, and suppressing the sensitivity of the prediction of the future state to the current local error. As shown in Figures 16 and 17, on average, the maximal eigenvalue of the Jacobian is smaller for the regularized model than for the basic model. Furthermore, the distribution of the eigenvalues of the Jacobian is shown in Figure 18 in the form of a linear stability diagram with explicit 5th order Runge-Kutta time integration. It is clear that the model with Jacobian regularization has significantly smaller positive real eigenvalues. Note that, due to the Frobenius norm, negative real eigenvalues are also decreased in magnitude.

4.5. Nonlinear PDE System: Instability-Driven Buoyant Mixing Flow. The test problems thus far have served to assess the performance of the basic and Jacobian regularized models on nonlinear dynamical systems that either evolve on or towards an attractor. Such systems, even if high-

dimensional, are amenable for projection onto a lower dimensional subspace, using, for instance, POD techniques. In this section, we consider the Boussinesq buoyant mixing flow [49, 50], also known as the unsteady lock-exchange problem [51] which exhibits strong shear and Kelvin-Helmholtz instability phenomena driven by the temperature gradient. Compared to the cylinder flow that evolves on a low-dimensional attractor approaching a limit cycle, the Boussinesq flow is highly convective and instability driven. Consequently, such a system state cannot be represented by a compact set of POD modes from the spatial-temporal field of nondimensionalized velocity and temperature. Rather, the low-dimensional manifold itself evolves with time. Further, any noise in the initial data can produce unexpected deviations that makes such systems challenging to model, even using equation-driven reduced order models such as POD-Galerkin [51].

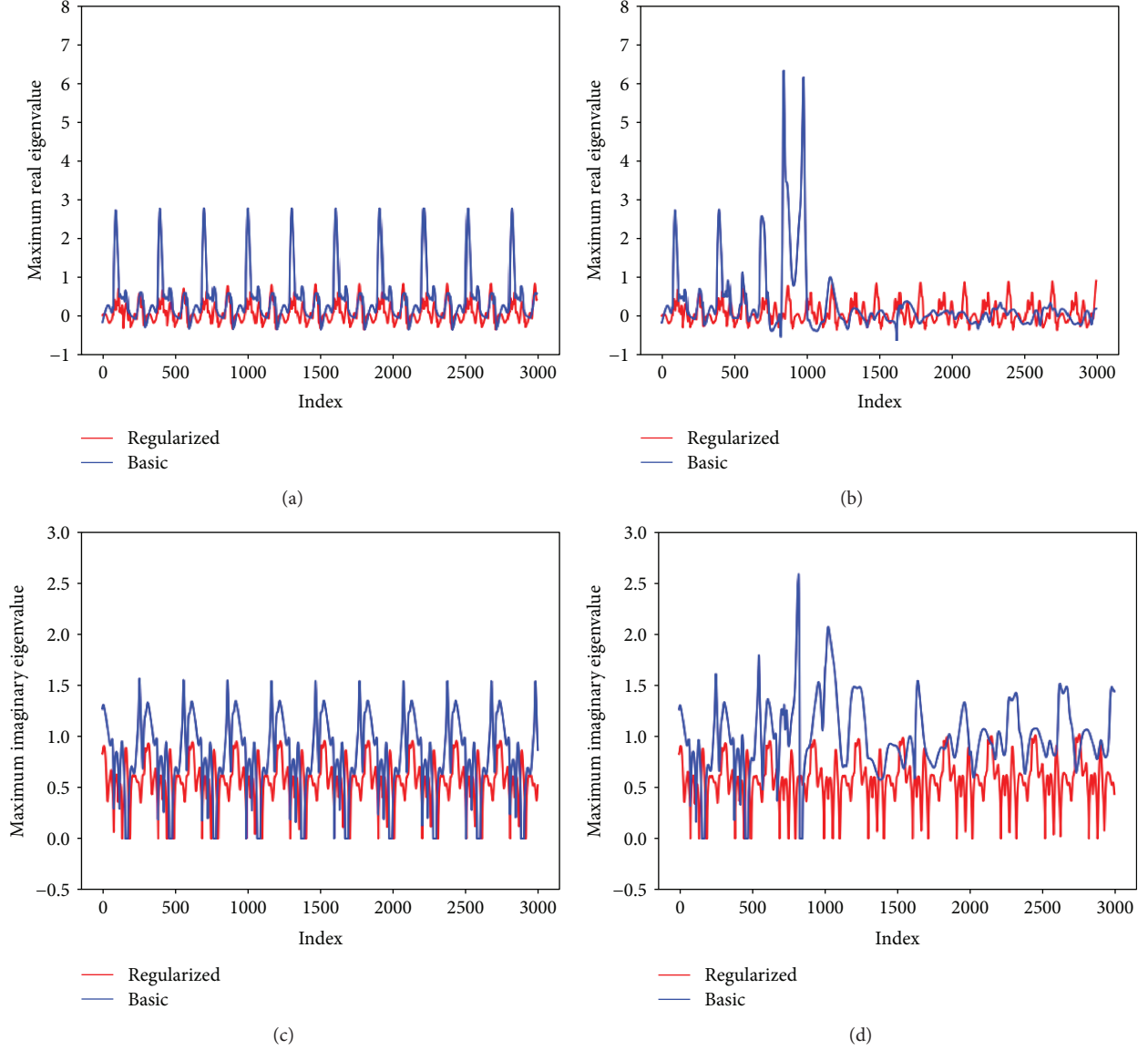


FIGURE 17: Flow in a cylinder wake. Comparison of eigenvalue of the Jacobian between regularized and basic model on testing data. (a, b) Maximal real eigenvalue. (c, d) Maximal imaginary eigenvalue. (a, c) A priori. (b, d) A posteriori.

The data set is generated by solving the dimensionless form of the two-dimensional incompressible Boussinesq equations [51], as shown in equation (30) on a rectangular domain that is $0 < x < 8$ and $0 < y < 1$.

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad (30a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \frac{1}{\text{Re}} \nabla^2 u, \quad (30b)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \frac{1}{\text{Re}} \nabla^2 v + \text{Ri} \theta, \quad (30c)$$

$$\frac{\partial \theta}{\partial t} + u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} = \frac{1}{\text{Re Pr}} \nabla^2 \theta, \quad (30d)$$

where u , v , and θ are the horizontal, vertical velocity, and temperature components, respectively. The dimensionless parameters Re , Ri , and Pr are the Reynolds number, Richardson number, and Prandtl number, respectively, with values chosen as follows: $\text{Re} = 1000$, $\text{Ri} = 4.0$, and $\text{Pr} = 1.0$. These equations are discretized on a 256×33 grid. Initially, fluids at two different temperatures are separated by a vertical line at $x = 4$. The bounding walls are treated as adiabatic with the no-slip condition. A fourth-order compact finite difference scheme is used to compute the derivatives in equation (30). The evolution of the thermal field over the simulation time interval of 32 seconds is shown in Figure 19 and illustrates the highly transient nature of the dynamics. To reduce the dimensionality of the system, POD modes are extracted from the entire data set consisting of 1600 snapshots. The reduced feature set consisting of ten POD weights captures nearly 97%

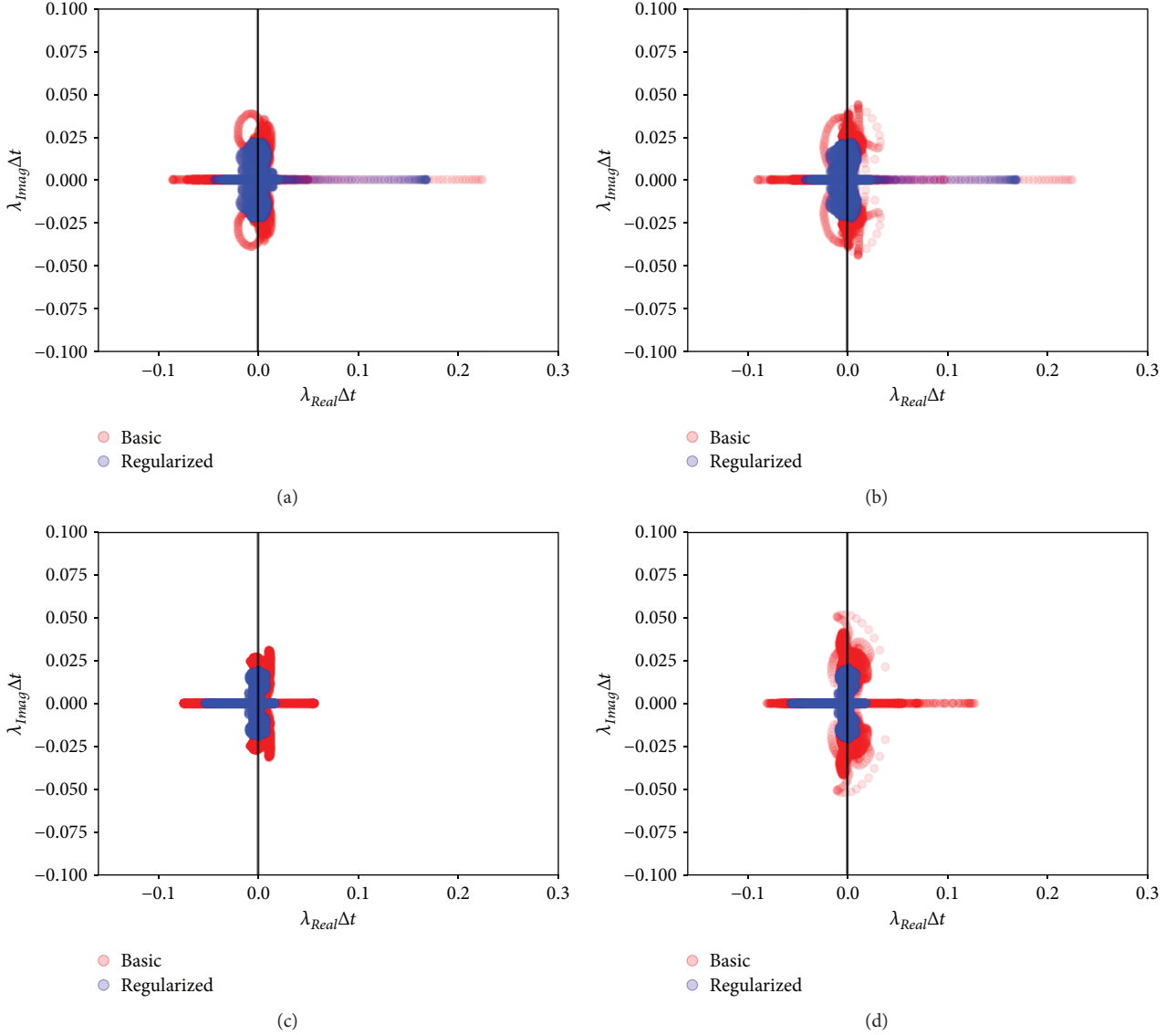


FIGURE 18: Flow in a cylinder wake. Comparison of linear stability diagram between regularized and basic model. (a) A priori on training data. (b) A posteriori on training data. (c) A priori on testing data. (d) A posteriori on testing data.

of the total energy is used to train the model and predict the trajectory.

For the setup of training and testing, future state prediction is pursued with the first 70% states of the trajectory treated as training data and the rest for testing. For such a system in 10 dimensions, it is observed that the problem of the a posteriori instability in the basic model becomes more pronounced and difficult to avoid. Challenges of numerical instability were observed even for reconstruction for a wide range of network configurations, and thus, results from the basic model are not reported.

The Jacobian regularized model is employed with hyperparameters shown in Table 4, with Figure 20 showing a posteriori evaluation on training data. The reconstruction is successful, but the performance deteriorates on testing data because the trajectory of the system does not exhibit a low dimensional attractor as in the cylinder case. Therefore, the training data is not informative for predictions on the test

set. For a black-box machine learning model, this phenomena can be expected to be more pronounced in high dimensional space due to data scarcity. Specifically, we discuss this problem in the following section.

4.6. Improving Model Predictability by Data Augmentation.

In this section, we consider two scenarios of data augmentation: (i) augmenting the information in the data by spreading training locations randomly following a uniform distribution provided that one has access to F_c or F_d at any desired location; (ii) augmenting the data by assembling several trajectories generated from different initial conditions.

4.6.1. Random Uniform Sampling in Phase Space. Recall that, in the two-dimensional problems in Section 4.1 and Section 4.2, the stepwise error contour shows that local error increases on testing scenarios located far away from the training data which was highly concentrated in a compact region

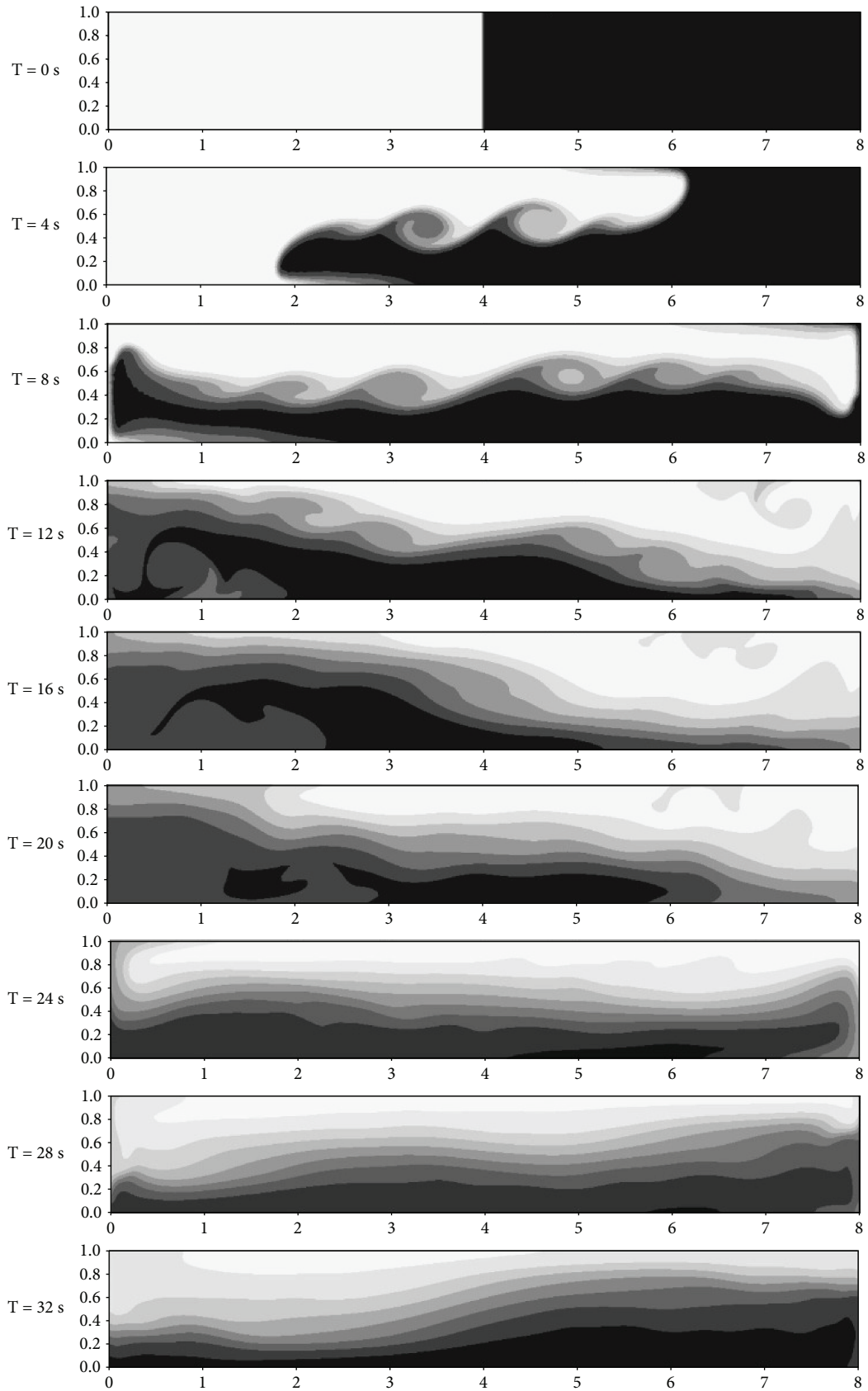


FIGURE 19: Time evolution of the temperature field for a two-dimensional Buoyant mixing flow.

TABLE 4: Hyperparameter configuration of the Jacobian regularized model for buoyant mixing flow.

Layer structure	Activation function	Loss function	Optimizer	Learning rate	λ
10-20-20-10	Penalized tanh [56]	MSE	Adam	0.001	5e-4

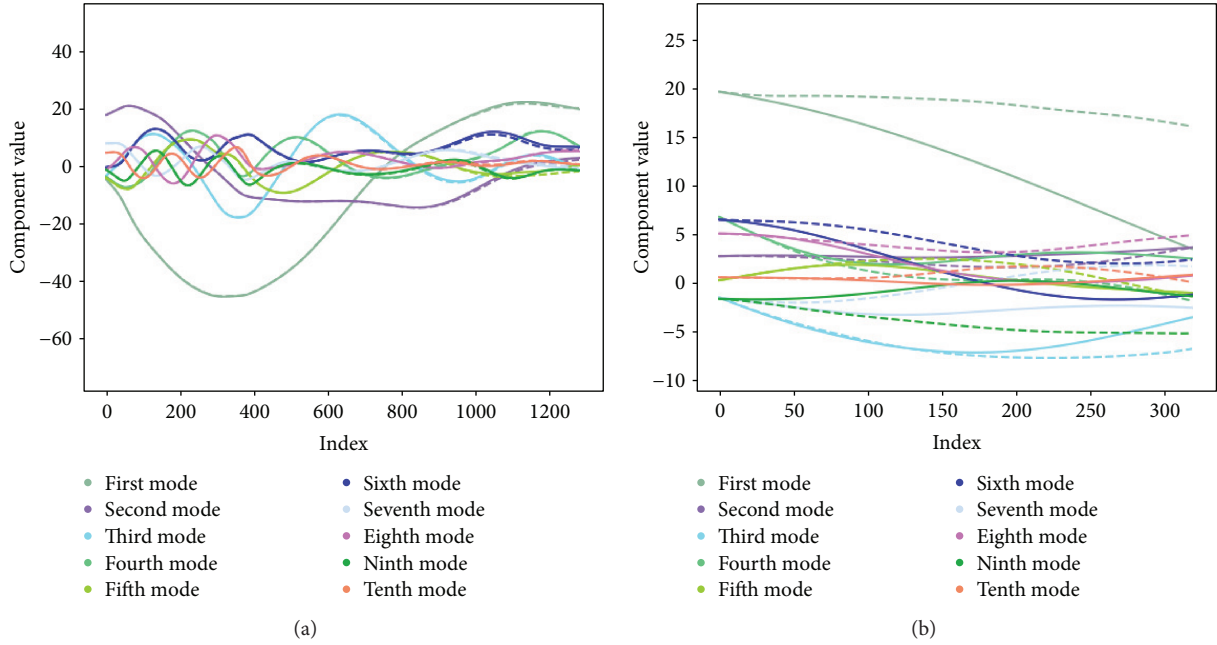


FIGURE 20: A posteriori comparison between prediction of Jacobian regularized model and ground truth for POD coefficient of buoyant mixing flow. Dashed: model. Solid: ground truth.

of phase space. Without any knowledge of system behavior, it is sensible to start with training data from a random uniform distribution in a compact region of phase space corresponding to interesting dynamics. To conduct a thorough stepwise error contour evaluation of the training target in phase space, the VDP system is chosen to illustrate this idea.

Determining the most informative data samples would potentially involve specific knowledge of the underlying system and the models used and is beyond the scope of the current work. Here, we simply consider uniform random sampling in the phase space in a finite domain: $[-3,3]$ for the first component and $[-5,5]$ for the second component. We obtained a new set of 399 training data points using random uniform sampling in phase space while retaining the same testing data as in Section 4.1.

The performance of the basic model with the same hyperparameter setting as in Section 4.1 on randomly distributed training data is shown in Figure 21. While the number of data points has not been changed, the contour error of the resulting model decreased significantly compared to training with the same number of data points in a single trajectory, which indicates an improved generalizability with the same amount of training data.

4.6.2. Training with Multiple Trajectories with Random Initialization. Training data can also be augmented by multiple trajectories with different initial conditions. Here, we take

the one-dimensional viscous Burgers equation shown in equation (31) as example.

The initial conditions are generated following a specific energy spectrum [28, 52] shown in equation (32).

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (31)$$

where $x \in [0, 2\pi]$ is a periodic domain discretized using 2048 uniformly distributed grid points, and $t \in [0, 20]$, $\nu = 0.01$.

$$u(x, 0) = \sum_{k=1}^{k_c} \frac{1}{\pi} \sqrt{2AE(k)} \sin(kx + \beta_k), \quad (32)$$

where for each k , β_k is a random number drawn from a uniform distribution on $[-\pi, \pi]$, $E(k) = 5^{-5/3}$ if $1 \leq k \leq 5$, $A = 25$, and $E(k) = k^{-5/3}$ if $k > 5$. Multiple trajectories are generated using different seeds for random numbers to obtain the trajectories of the full-order system. To fully resolve the system as a DNS, equation (31) is solved using a standard pseudospectral method with SSP-RK3 [53] for time stepping. Here, we choose $k_c = 2$. Discrete cosine transformation (DCT) is used to reduce the dimension of the full system to first 4 cosine modes in the system where around 97% of kinetic energy is preserved. For simplicity, we seek a closed Markovian

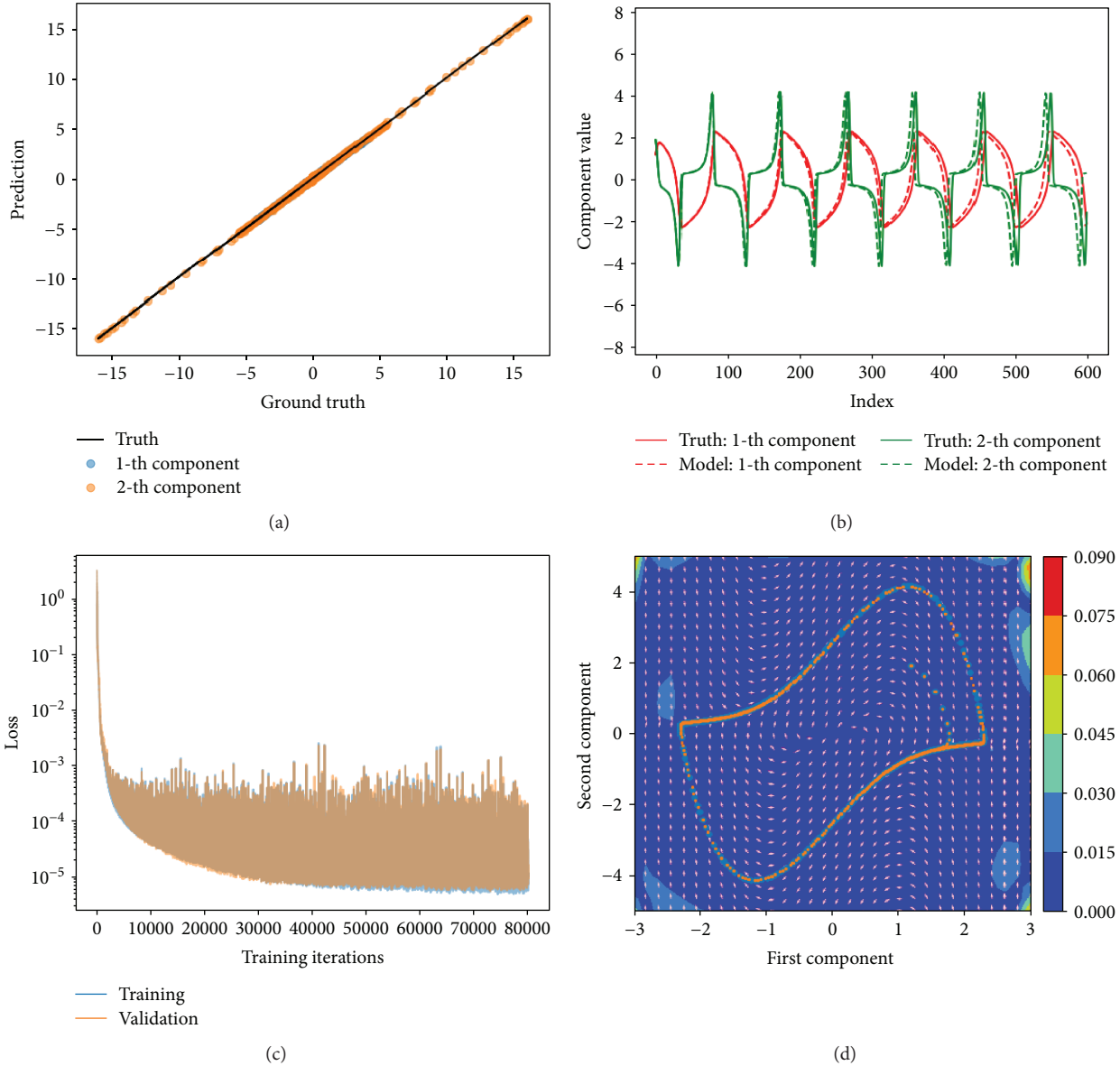


FIGURE 21: A priori and a posteriori result of the basic model on the VDP system, training with random sampled data distribution. (a) A priori evaluation. (b) A posteriori evaluation. (c) Learning curve. (d) Stepwise error contour. Blue dot: ground truth. Orange dot: prediction. White arrow: direction of target vector of ground truth. Red arrow: direction of target vector of prediction.

reduced-order-system, whereas the underlying dynamics is clearly non-Markovian [28, 54].

Since the first component of the DCT is constant, the remaining components of feature space are shown in Figure 22. The training data is far away from the testing data initially, whereas the data converges at a later stage. This is because of the presence of a spiral fixed point attractor resulting from the viscous dissipative nature of the system. Therefore, if the model is only trained from a single trajectory, it will be very difficult for the model to generalize well in the phase space especially where the state of the system is not near an attractor.

Many dynamical systems in nature exhibit attractors in the asymptotic sense. From the viewpoint of data-driven modeling of such dynamics, data scarcity is encountered

at the start of trajectory where the number of trajectories required to provide enough information to cover the region of interest grows exponentially. Much research on applying neural network-based models for dynamical systems [16, 20, 34] demonstrate problems starting on limit cycles or chaotic attractors in a low-dimensional feature space, where the issue of initial data scarcity is not significant, or can be easily alleviated by a small increase in available data. However, for the purpose of modeling phenomena such as turbulent fluid flow, which can be high dimensional even after dimension reduction, the model would likely fail for long-time prediction due to data scarcity. Such a situation may be realized in regions of phase space where the state has not arrived at the low manifold attractor. Therefore, the training data might not be representative of testing

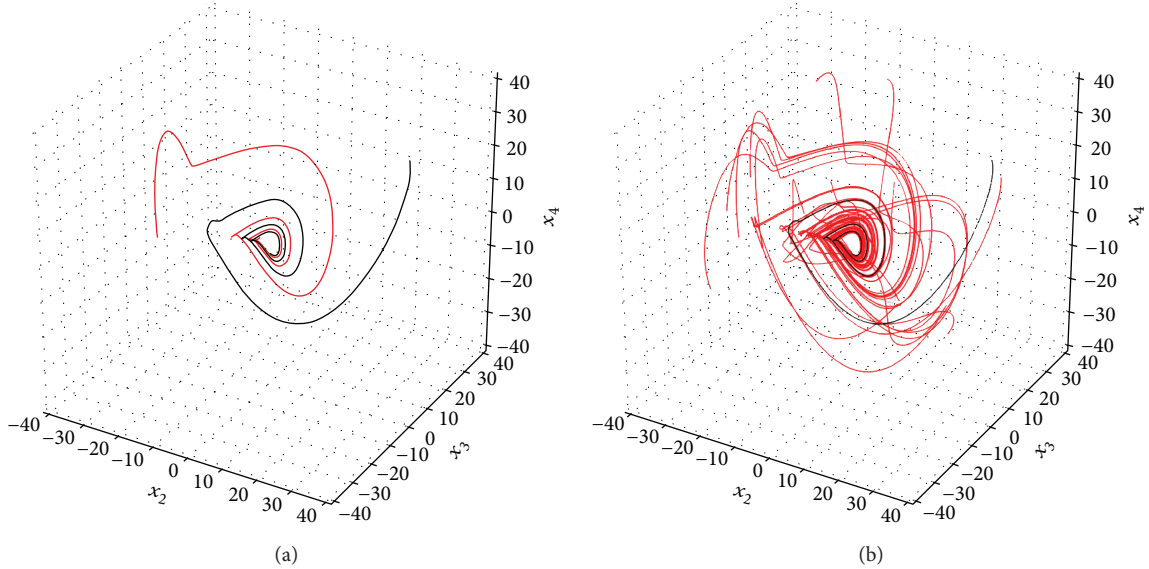


FIGURE 22: Feature distribution of training/testing data for component x_2, x_3, x_4 . (a) With one training trajectory. (b) With 18 training trajectory. Black line: testing data. Red line: training data.

TABLE 5: Hyperparameter configuration of the basic model: VBE system with four modes.

Layer structure	Activation function	Loss function	Optimizer	Learning rate
4-30-30-30-4	elu	MSE	Adam	0.0005

data which violates the fundamental assumption of a well-posed machine learning problem [55]. Moreover, data scarcity will shrink the region of generalizability of the model as the dimension of the system increases.

A key benefit of using a neural network model is its linear growth in complexity with dimension of the system, in contrast to traditional polynomial regression methods [35]. However, initial data scarcity would limit the generalizability of an ANN in modeling a high dimensional dynamical system that does not exhibit a low dimensional attractor. We believe that this phenomenon of data scarcity observed from this simple nonlinear PDE example also applies to other nonlinear dynamical systems.

To alleviate the initial data scarcity issue, a solution is to augment the training data with more trajectories with different random number seeds in generating the initial condition, while keeping the energy spectrum the same across all cases. In this case, we choose 18 such trajectories. Each trajectory contains states of 1000 snapshots equally spaced in time. For testing data, we simply consider one DNS result with an initial condition different from all training trajectories. The corresponding training and testing trajectories are visualized in phase space as shown in Figure 22.

The basic model is trained with hyperparameters in Table 5 and 1000 epochs. The resulting learning curve and a posteriori evaluation are shown in Figure 23. Relatively large discrepancy is observed near the initial condition as the initial data scarcity is not completely eliminated due to limited number of additional trajectories. Increasing the number of additional trajectories may be unaffordable for

very high dimensional systems. Moreover, the result also shows that the error decreases once the trajectory falls on the fixed point attractor. Thus, if the model starts in the low dimensional attractor where the information is well-preserved in the training data, better performance might be expected. This hypothesis is consistent with the previous work [34], where successful prediction of future states starts at the time when the states converge to a low dimensional attractor.

5. Conclusions

This work investigated the modeling of dynamical systems using feedforward neural networks (FNN), with a focus on long time prediction. It was shown that neural networks have advantages over sparse polynomial regression in terms of adaptability, but with a trade-off in training cost and difficulty in extrapolation, which is a natural barrier for almost all supervised learning. From the perspective of global error analysis, and the observation of the strong correlation between the local error and maximal singular value of the Jacobian, we propose the suppression of the Frobenius norm of the Jacobian as regularization. This showed promise in improving the robustness of the basic FNN model given limited data, or when the model has a nonideal architecture, or when the model is unstable. The effectiveness of Jacobian regularization is attributed to finding a balance between lowering the prediction error and suppressing the sensitivity of the prediction of the future state to the current local error. In terms of modeling dynamical systems that do not involve

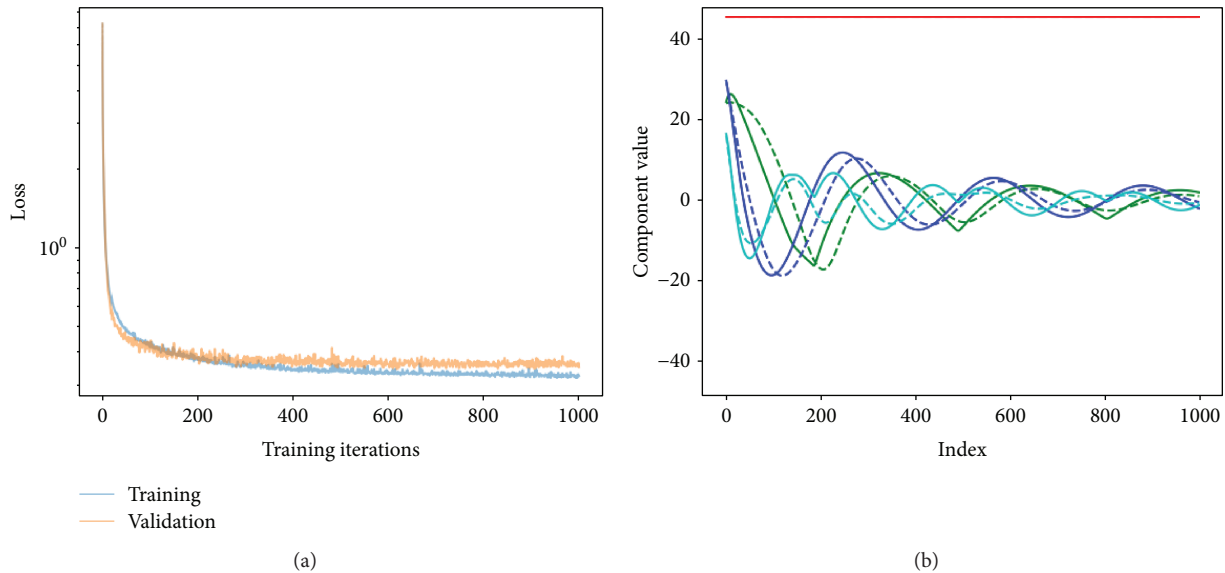


FIGURE 23: Result of the basic model for one-dimensional VBE turbulence case. (a) Learning curve. (b) A posteriori result with augmented training trajectories. Red: x_1 . Green: x_2 . Blue: x_3 . Cyan: x_4 . Dashed: model prediction. Solid: ground truth.

low-dimensional attractors, limitations of FNNs, and perhaps all local ML methods, was demonstrated in a buoyant mixing flow. Challenges were noted in the example of the reduced-order viscous burgers system, where significant initial data scarcity is present. Augmenting the data either by altering the distribution of training data in phase space or by simply adding multiple trajectories from different initial conditions resulted in improvement of the performance of FNN model to some extent. However, these remedies require a significant amount of additional sampling in phase space, especially for high dimensional systems for the period of time without the apparent low-dimensional attractor which suffers data sparsity from the curse of dimensionality.

Data Availability

Most of the data contained in the current study are synthetic data which can be easily generated through the configuration described. In addition, the cylinder flow data is directly downloadable in the public repo of SINDy [41]: faculty.washington.edu/sbrunton/sparsedynamics.zip. And the instability-driven buoyant mixing flow data were supplied by Professor Balaji Jayaraman under license and so cannot be made freely available. Requests for access to these data should be made to balaji.jayaraman@okstate.edu. Furthermore, all of the data used to support the findings of this study are available from the corresponding author upon request. Ref: [1] Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems." *Proceedings of the National Academy of Sciences* 113.15 (2016): 3932–3937.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge Professor Balaji Jayaraman for his helpful discussion and comments and for providing the data for the two-dimensional instability-driven Buoyant mixing flow. This work was supported by AFOSR and AFRL under grants FA9550-16-1-0309 and FA9550-17-1-0195.

References

- [1] M. Tanaskovic, L. Fagiano, C. Novara, and M. Morari, "Data-driven control of nonlinear systems: an on-line direct approach," *Automatica*, vol. 75, pp. 1–10, 2017.
- [2] R. H. Shumway and D. S. Stoffer, "Time series analysis and its applications," *Studies in Informatics and Control*, vol. 9, no. 4, pp. 375–376, 2000.
- [3] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [4] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, 2016.
- [5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Largescale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, Columbus, OH, USA, 2014.
- [6] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [7] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*, Springer, 2017.

- [8] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [9] J. G. Kuschewski, S. Hui, and S. H. Zak, "Application of feedforward neural networks to dynamical system identification and control," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 1, pp. 37–49, 1993.
- [10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [11] K. S. Narendra and K. Parthasarathy, "Neural networks and dynamical systems," *International Journal of Approximate Reasoning*, vol. 6, no. 2, pp. 109–131, 1992.
- [12] M. M. Polycarpou and P. A. Ioannou, "Identification and control of nonlinear systems using neural network models: design and stability analysis," University of Southern California, 1991.
- [13] S. A. Billings, H. B. Jamaluddin, and S. Chen, "Properties of neural networks with applications to modelling nonlinear dynamical systems," *International Journal of Control*, vol. 55, no. 1, pp. 193–224, 1992.
- [14] V. T. S. Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594–603, 1994.
- [15] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural Networks*, vol. 5, no. 6, pp. 961–970, 1992.
- [16] F.-S. Tsung and G. W. Cottrell, "Phase-space learning," in *Advances in Neural Information Processing Systems*, pp. 481–488, MIT Press, 1995.
- [17] T. L. Paez and N. F. Hunter, "Dynamical system modeling via signal reduction and neural network simulation," Sandia National Labs, Albuquerque, NM (United States), 1997.
- [18] T. L. Paez and N. F. Hunter, "Nonlinear system modeling based on experimental data," Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.
- [19] A. Urbina, N. F. Hunter, and T. L. Paez, "Characterization of nonlinear dynamic systems using artificial neural networks," Technical report, Sandia National Labs, Albuquerque, NM (United States), 1998.
- [20] N. Smaoui, "Artificial neural network-based low-dimensional model for spatio-temporally varying cellular flames," *Applied Mathematical Modelling*, vol. 21, no. 12, pp. 739–748, 1997.
- [21] N. Smaoui, "A model for the unstable manifold of the bursting behavior in the 2D navier–stokes flow," *SIAM Journal on Scientific Computing*, vol. 23, no. 3, pp. 824–839, 2001.
- [22] N. Smaoui and S. Al-Enezi, "Modelling the dynamics of nonlinear partial differential equations using neural networks," *Journal of Computational and Applied Mathematics*, vol. 170, no. 1, pp. 27–58, 2004.
- [23] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [24] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [25] R. Bakker, J. C. Schouten, C. L. Giles, F. Takens, and C. M. van den Bleek, "Learning chaotic attractors by neural networks," *Neural Computation*, vol. 12, no. 10, pp. 2355–2383, 2000.
- [26] H. Y. Lin, W. Chen, and A. Tsutsumi, "Long-term prediction of nonlinear hydrodynamics in bubble columns by using artificial neural networks," *Chemical Engineering and Processing: Process Intensification*, vol. 42, no. 8–9, pp. 611–620, 2003.
- [27] A. Chorin and O. H. Hald, *Stochastic Tools in Mathematics and Science*, vol. 3 of Surveys and Tutorials in the Applied Mathematical Sciences, Springer, 2009.
- [28] E. Parish and K. Duraisamy, "Reduced order modeling of turbulent flows using statistical coarse-graining," in *46th AIAA Fluid Dynamics Conference*, Washington, D.C., USA, 2016.
- [29] T. Koskela, M. Lehtokangas, J. Saarinen, and K. Kaski, "Time series prediction with multilayer perceptron, FIR and Elman neural networks," in *Proceedings of the World Congress on Neural Networks*, pp. 491–496, 1996, Citeseer.
- [30] D. S. Broomhead and G. P. King, "Extracting qualitative dynamics from experimental data," *Physica D: Nonlinear Phenomena*, vol. 20, no. 2–3, pp. 217–236, 1986.
- [31] T. Miyoshi, H. Ichihashi, S. Okamoto, and T. Hayakawa, "Learning chaotic dynamics in recurrent RBF network," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 1, pp. 588–593, Perth, WA, Australia, 1995.
- [32] Y. Sato and S. Nagaya, "Evolutionary algorithms that generate recurrent neural networks for learning chaos dynamics," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 144–149, Nagoya, Japan, 1996.
- [33] C. A. L. Bailer-Jones, D. J. C. MacKay, and P. J. Withers, "A recurrent neural network for modelling dynamical systems," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. 531–547, 1998.
- [34] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, "Model identification of reduced order fluid dynamics systems using deep learning," *International Journal for Numerical Methods in Fluids*, vol. 86, no. 4, pp. 255–268, 2018.
- [35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.
- [36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [37] A. P. Trischler and G. M. T. D'Eleuterio, "Synthesis of recurrent neural networks for dynamical system simulation," *Neural Networks*, vol. 80, pp. 67–78, 2016.
- [38] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [39] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, <https://arxiv.org/abs/1412.6980>.
- [40] M. Abadi, A. Agarwal, P. Barham et al., "Large-scale machine learning on heterogeneous distributed systems," CoRR, 2016, <https://arxiv.org/abs/1603.04467>.
- [41] J. Bergstra and Y. Bengio, "Random search for hyperparameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [42] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proceedings 30th International Conference on Machine Learning*, pp. 115–123, 2013.
- [43] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, pp. 2951–2959, NIPS Foundation, 2012.

- [44] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent, "Adding noise to the input of a model trained with a regularized objective," 2011, <https://arxiv.org/abs/1104.3250>.
- [45] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," CoRR, 2017, <https://arxiv.org/abs/1710.05941>.
- [46] R. Yu, S. Zheng, and Y. Liu, "Learning chaotic dynamics using tensor recurrent neural networks," in *Proceedings of the ICML 17 workshop on deep structured prediction*, vol. 70, Sydney, Australia, 2017PMLR.
- [47] K. Taira and T. Colonius, "The immersed boundary method: a projection approach," *Journal of Computational Physics*, vol. 225, no. 2, pp. 2118–2137, 2007.
- [48] B. R. Noack, K. Afanasiev, M. A. R. E. K. MORZYNSKI, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *Journal of Fluid Mechanics*, vol. 497, pp. 335–363, 2003.
- [49] J.-G. Liu, C. Wang, and H. Johnston, "A fourth order scheme for incompressible Boussinesq equations," *Journal of Scientific Computing*, vol. 18, no. 2, pp. 253–285, 2003.
- [50] E. Weinan and C.-w. Shu, "Small-scale structures in Boussinesq convection," *Physics of Fluids*, vol. 6, no. 1, pp. 49–58, 1994.
- [51] O. San and J. Borggaard, "Principal interval decomposition framework for pod reduced-order modeling of convective Boussinesq flows," *International Journal for Numerical Methods in Fluids*, vol. 78, no. 1, pp. 37–62, 2015.
- [52] Y. Li and Z. J. Wang, "A priori and a posteriori evaluations of sub-grid scale models for the burgers equation," *Computers & Fluids*, vol. 139, pp. 92–104, 2016.
- [53] S. Gottlieb, C.-W. Shu, and E. Tadmor, "Strong stability-preserving high-order time discretization methods," *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001.
- [54] S. Pan and K. Duraisamy, "Data-driven discovery of closure models," 2018, <https://arxiv.org/abs/1803.09318>.
- [55] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning, Volume 1*, Springer series in statistics New York, Springer-Verlag, New York, NY, USA, 2001.
- [56] B. Xu, R. Huang, and M. Li, "Revise saturated activation functions," 2016, <https://arxiv.org/abs/1602.05980>.

Research Article

A Damage Classification Approach for Structural Health Monitoring Using Machine Learning

Diego Tibaduiza ¹, Miguel Ángel Torres-Arredondo,² Jaime Vitola,^{3,4}
Maribel Anaya,⁵ and Francesc Pozo ³

¹Departamento de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Colombia, Cra 45 No. 26-85, Bogotá, Colombia

²MAN Energy Solutions SE, Test & Validation—R&D Engineering Four-Stroke (EEFTTM), Stadtbachstr. 1, 86153, Augsburg, Germany

³Control, Modeling, Identification, and Applications (CoDALab), Departament de Matemàtiques, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besòs (CDB), Eduard Maristany, 16, Barcelona 08019, Spain

⁴MEM (Modelling, Electronics, and Monitoring Research Group), Faculty of Electronics Engineering, Universidad Santo Tomás, Cra. 9 Nos. 51-11 Bogotá, Colombia

⁵Faculty of Electronics Engineering, Universidad Sergio Arboleda, Calle 74 #14-14, Bogotá, Colombia

Correspondence should be addressed to Francesc Pozo; francesc.pozo@upc.edu

Received 6 July 2018; Revised 3 October 2018; Accepted 31 October 2018; Published 2 December 2018

Academic Editor: Francisco Chinesta

Copyright © 2018 Diego Tibaduiza et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Inspection strategies with guided wave-based approaches give to structural health monitoring (SHM) applications several advantages, among them, the possibility of the use of real data from the structure which enables continuous monitoring and online damage identification. These kinds of inspection strategies are based on the fact that these waves can propagate over relatively long distances and are able to interact sensitively with and uniquely with different types of defects. The principal goal for SHM is oriented to the development of efficient methodologies to process these data and provide results associated with the different levels of the damage identification process. As a contribution, this work presents a damage detection and classification methodology which includes the use of data collected from a structure under different structural states by means of a piezoelectric sensor network taking advantage of the use of guided waves, hierarchical nonlinear principal component analysis (h-NLPCA), and machine learning. The methodology is evaluated and tested in two structures: (i) a carbon fibre reinforced polymer (CFRP) sandwich structure with some damages on the multilayered composite sandwich structure and (ii) a CFRP composite plate. Damages in the structures were intentionally produced to simulate different damage mechanisms, that is, delamination and cracking of the skin.

1. Introduction

Data-driven algorithms have demonstrated their utility in structural health monitoring (SHM) applications. In fact, the use of this kind of approaches is a useful tool for real-time condition monitoring (CM). However, one of the challenges in the use of data-driven algorithms is associated with the size and quantity of information which is often obtained from sensor networks or multiple sensors. This information represents a great deal of data to process and analyse. In this sense, it is necessary to develop better methodologies which allow avoiding false alarms in the damage identification process.

An SHM system typically includes five steps in its design: these are (i) sensor network design; (ii) data acquisition; (iii) feature extraction, (iv) diagnosis, and (v) prognosis. The first four stages normally involve methods for data-sensor fusion, multivariate statistical modelling and pattern recognition algorithms. For the later, a physics-based model is almost inevitable so that reliable predictions can be performed. It is evident that structural health monitoring systems have been advancing worldwide as shown by the amount of relevant available scientific papers and recent practical applications [1–3]. Among the solutions in the application of data driven algorithms for SHM, there are many applications in bridges

[4–6], aeronautics [7, 8], aerospace [9, 10], wind turbines [11–13], among others.

As a contribution to the development of new ways to process and evaluate the condition of a structure using data from sensors, a methodology for damage classification and detection is presented in this paper. This work is also motivated by the need to further develop, integrate and evaluate damage identification algorithms [7, 14, 15]. The proposed methodology is based on an acousto-ultrasonic approach in which ultrasonic waves are generated in a piezoelectric transducer sensor network in several actuation phases. The captured signals are preprocessed by means of the discrete wavelet transform (DWT) for feature extraction and then integrated into a nonlinear multivariate model where some nonlinear components are generated in order to form feature vectors for all the actuation phases and to train a machine by means of the machine learning point of view. Afterward, measurements with the sensor network are captured from the structure in an unknown state and the interaction with the training machine allows defining the current structural state according to the states defined in the training step. To validate the proposed methodology, experiments are carried out in a composite sandwich structure in which increasing damage is intentionally introduced and a composite plate with simulated damages.

The remaining part of this paper is organized as follows. For completeness, the article first presents a brief summary of the basic theoretical background for the different evaluated signal processing algorithms. Afterward, the methodology is introduced in Section 3. Section 4 is devoted to the experimental validation, where the experimental setup and results are included. Finally, conclusions are given in Section 5.

2. Theoretical Background

This section introduces some brief concepts about some well-known methods that are used in the developed methodology. Authors suggest reviewing the references in each subsection if more information about each method is required.

2.1. Discrete Wavelet Transform. The discrete wavelet transform (DWT) is a very useful tool, used in an increasingly broad horizon, image processing, health care, energy distribution, SHM, and others. That can be defined as a filter bank structure to distinguish features through the use of low-pass filters and high-pass filters [16, 17]. This configuration allows representing the variability of a given function by means of coefficients at a specified time and scale. These coefficients are calculated by using quadrature mirror filters and are decomposed in approximation (A1, A2, . . .) and detail coefficients (D1, D2, . . .) [7] as is shown in Figure 1.

Detail coefficients are low-scale, high-frequency components, while the approximation coefficients represent the high-scale, low-frequency components. The wavelet technique has been of great interest in recent years and has direct application for the SHM like demonstrates some research works [18–22]. For further details about DWT and its implementation, please refer to [23].

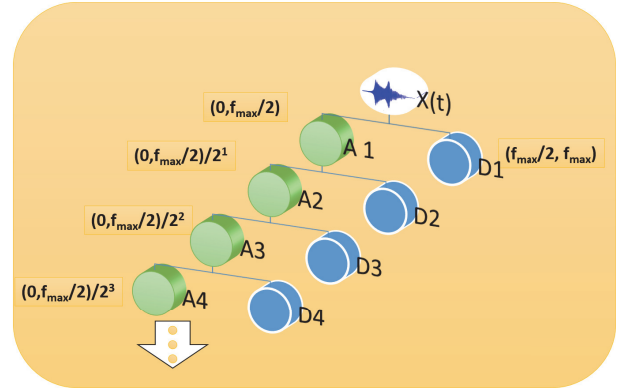


FIGURE 1: Discrete wavelet transform decomposition.

2.2. Hierarchical Nonlinear Principal Component Analysis.

The hierarchical nonlinear principal component analysis is also known as h-NLPCA and is also defined as a nonlinear generalization of traditional PCA [24]. This is a method based on a multi-layered perceptron (MLP) architecture with an auto-associative topology. The auto-associative network works with the inputs and outputs to perform the identity mapping by using the square error [24]. This architecture, shown in Figure 2, includes a bottleneck layer which allows us to compress data and reduce the dimension of the original data. Note that the nodes in the mapping and demapping layers must have nonlinear transfer functions; nonlinear transfer functions are not necessary for the bottleneck layer [25]. With the purpose of guaranteeing that the calculated nonlinear components have the same hierarchical order as the linear components in standard principal component analysis (PCA) and in contrast to standard NLPCA, the reconstruction error is controlled by searching a k dimensional subspace of minimum mean square error (MSE) under the constraint that the $(k-1)$ dimensional subspace is also of minimal MSE [26].

This process is repeated for any k -dimensional subspace where all subspaces must be of minimal MSE. h-NLPCA describes the data with greater accuracy and/or by fewer factors than PCA, provided that there are sufficient data to support the formulation of more complex mapping functions [27, 28].

2.3. Machine Learning. In the recent years, the machine learning (ML) has been the focus of many researchers in the area of structural health monitoring (SHM) by its effectiveness and continuous development [29–32]. Machine learning is a set of algorithms that can extract, in an automatic way, the hidden patterns in a large group of data [33, 34]. There are two different approaches in ML according to the training process:

- (i) *Supervised*, where the machine gets the inputs and the expected outputs. The machine is trained to find the complex patterns and relationships between them and obtain generalized responses based on this training with right answers [35, 36].

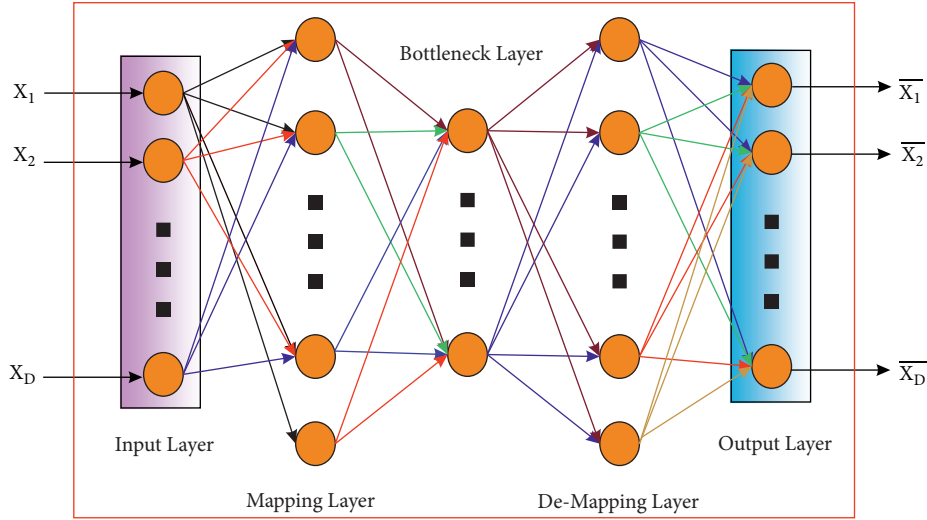


FIGURE 2: Network architecture for h-NLPCA.

- (ii) *Unsupervised*, where the machine is trained to find the similarities in the data and provide a clustering organization to indicate its proximity [37, 38].

In this work, a supervised training is explored; in this sense, some of the supervised machines used in the methodology are then explained. On one hand, k nearest neighbours (k NN) is a machine learning algorithm that has a very simple strategy. More precisely, the elements are classified by the distance to others and the frequency with which this proximity is presented. It is important not to take the risk of overfitting. In this case, the trained machine will only apply for the current group of data. Therefore, to ensure that this does not occur, it is important to keep a moderate number of characteristics and training examples. On the other hand, decision trees are a predictive model used in, for instance, data mining and statistics. This mechanism maps the observations in a structure that allows us to reveal conclusions about these observations. This structure also allows us to extrapolate these conclusions and predict new behaviours with new observations. To extract the desired structure that describes the information, an analysis of the data and the critical values that builds a better division of them is performed. This division is performed after locating the choice nodes and the change nodes in the decision structure with the aim of obtaining a better decision branches and a best behaviour in the prediction.

In order to facilitate that the machine reaches the goal, it is very common to simplify the input data through some techniques [2, 39]. In this work, only the supervised type is explored and results are presented by the use of the confusion matrices. These types of matrices are a very useful tool to classify data considering the following classes: true positives, false negatives, false positives and true negatives.

3. Damage Classification Methodology

In this work, piezoelectric transducers were used because these devices are cheap, easy to install, lightweight, and with

several other good characteristics [40, 41]. Figure 3 shows a representation of the methodology applied. This can be divided into two parts: training and testing, where in both cases the strategy uses data from the structure collected by a piezoelectric sensor network in several actuation phases. This network is built with several piezoelectric transducers which are attached to the structure under test in a permanent way and distributed over its surface as in Figures 9 and 13. Because these transducers can work as actuators or as sensors, each actuation phase is defined by a PZT working as an actuator and using the rest of PZTs as sensors, this procedure is repeated for each PZT in the sensor network [42]. This means that an excitation signal is applied to a piezoelectric sensor and propagated signals through the structure are collected by the rest of sensors, organized and preprocessed. This process is repeated for each sensor in the structure [43, 44]. Each signal captured by the acquisition system is preprocessed by the Discrete Wavelet Transform at a defined decomposition level and, as result, a reduced signal is obtained and organized by each actuation phase as in [43]. These steps are the same for training and testing steps. Once data are preprocessed and organized, during the training step, h-NLPCA is applied to the data by each actuation phase and a determined number of nonlinear components are obtained and used for training the machines; in particular for the explored cases in this paper the first three scores (S_1 , S_2 , and S_3) were used by each actuation phase to define the feature vector for training the machines as it is shown in Figure 4. This Figure is an example when only four sensors are used as in the case of the specimen 1. As result of this step, a machine with the information of the structural states is trained and is available for the testing step.

Testing is performed by using data from the structure in an unknown structural state and projecting the information to the nonlinear components, as results of this projection appear the scores which are used as input to the trained machine to predict the kind of structural state. This procedure allows us to detect and classify the structural state.

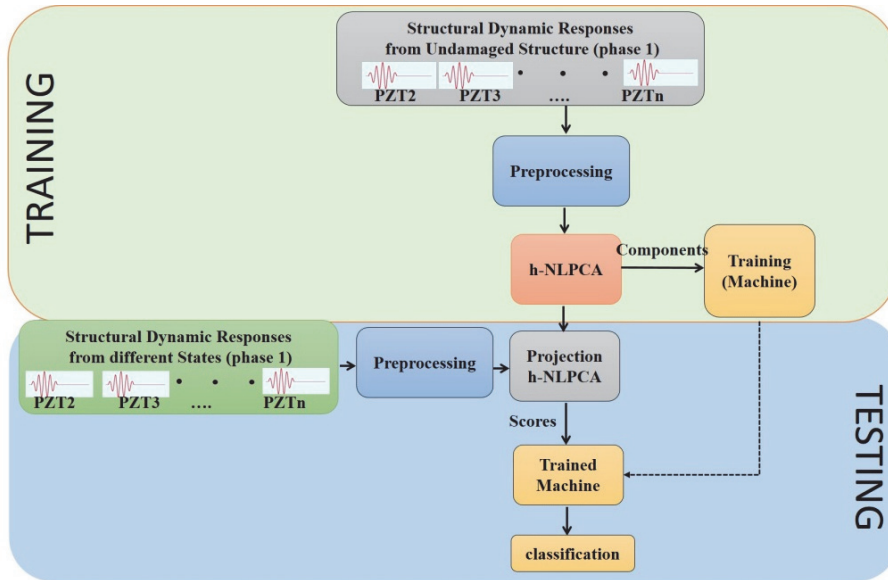


FIGURE 3: Damage classification methodology.

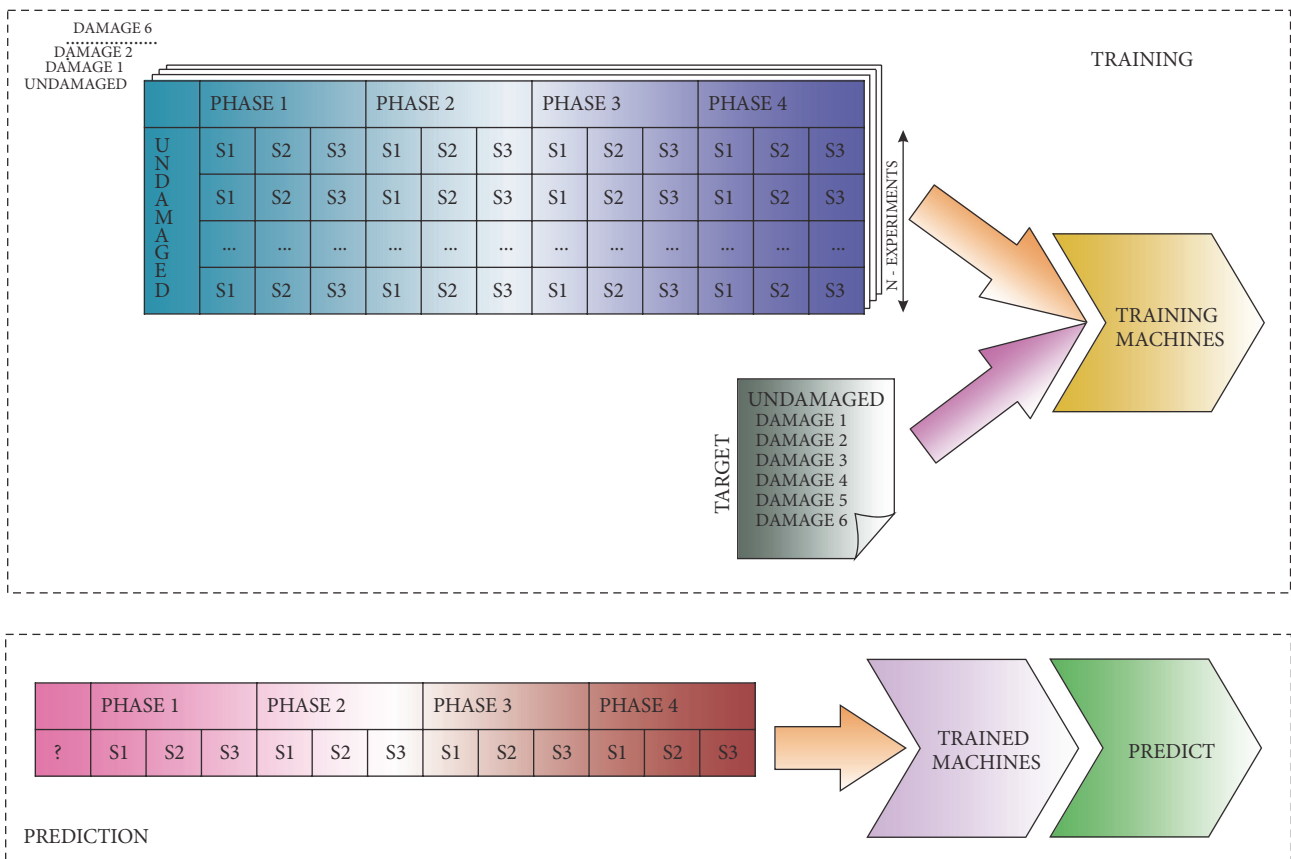


FIGURE 4: Methodology-machine learning.

4. Experimental Validation

To validate the methodology, data from two structures are considered. A carbon fiber-reinforced plastic (CFRP)

sandwich structure with some damages on the multilayered composite sandwich structure and a CFRP plate with an added mass to simulate damages were used. Several experiments were collected per each structural damage state

TABLE I: Damage description.

Damage Number	Description
1	Delamination: started symmetrically from the right side of the sample at its middle position along the y-axis. Its width along the y-axis is 16 mm and its depth along the x-axis is 10 mm
2	Extended the previous damage to a width of 33 mm and depth of 42 mm
3	A crack of 25 mm length initiated at the middle position along the vertical y-axis and in the parallel direction to the x-axis
4	Extended the previous crack to a length of 30 mm
5	Extended the previous crack to a length of 45 mm
6	Extended the previous crack to a length of 70 mm

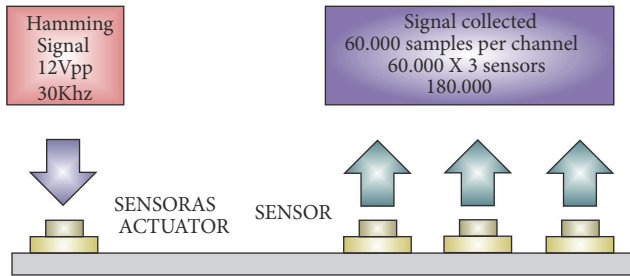


FIGURE 5: Structure exploration.

(including undamaged state) to train the machines and to test the behaviour of the prediction as will be explained in the following subsections. In addition, there is a detailed description of the measurement procedure, the structures, and the results obtained from the use of the developed methodology.

4.1. Measurement Procedure. As it has been previously introduced in the last section, the interaction with the structure is performed by the signals applied and collected to the piezoelectric sensor network. In the cases of the structures evaluated in this paper, piezoelectric sensors PIC-151 were used. The inspection is performed in four phases for the specimen 1 and nine phases for the specimen 2 due to the number of piezoelectric sensors installed in each structure. During the first actuation phase (phase I), the first piezoelectric was stimulated with a Hamming windowed cosine signal, 12 volts of the peak value, and a frequency determined for each structure, and the information of the interaction of the propagated waves with the structure is collected in different places of the structure by the rest of the sensors. Figure 5 describes an example of this actuation phase. The second actuation phase (phase II) implies the use of the second piezoelectric as an actuator and the rest used as sensors and so on. This process ends when all piezoelectric transducers have been used as an actuator. All this information is stored for the subsequent process in several matrices and files, one per each actuation phase.

Experiments consider different structural states (healthy and structure with damage in different positions) as it will be

explained in the following subsections. Number of samples of each sensor is 60.000. This means that the number of columns in this matrix is $(n-1)$ sensors \times 60.000 samples. Figure 6 shows this organization, the corresponding preprocessing, and the procedure to extract the h-NLPCA scores for the case of a structure with four piezoelectric sensors. In this case, the first 30 scores are retained during the model construction with h-NLPCA.

After scores are obtained per each actuation phase, the feature vector for training is defined. Figure 7 shows the assembled training vector. In this case, training is made with a vector of twelve elements (three scores from each actuation phase), this means that, for instance, in the case of specimen 1 with 4 sensors, 4 actuation phases were considered as in Figure 7. In the same way, the Figure shows the case where 150 experiments were acquired for each structural damage state. With respect to the normalization, group scaling was used in each matrix from each actuation phase [45].

These steps are repeated in the same way for data during training and testing process. Following some details about the particular experiments with each evaluated specimen will be presented.

4.2. Specimen 1: CFRP Sandwich Structure. The first structure corresponds to a CFRP sandwich structure (Figure 8), where the damages are intentionally produced to simulate different damage mechanisms, i.e. delamination and cracking of the skin. These damage mechanisms are summarized in Table 1. The overall size of this structure is 217 mm x 217 mm x 31 mm and it is made of carbon/epoxy material with a 0.5 mm thickness. The stacking sequence is $[0^\circ 90^\circ]$ (Figure 9).

The core is made of polyetherimide foam with a 30 mm thickness. Four PIC-151 piezoelectric transducers from PI Ceramics are attached to the surface of the structure with equidistant spacing. Figure 9 shows a photo of the experiment.

The scan frequency was a 50 kHz, with a peak voltage of 12 V, and Hamming windowed cosine form, with five cycles. Seven structural states were studied (healthy state and six damages) as it was previously explained. In each structural state 150 experiments were performed, according to the following distribution: 100 experiments were used for training and 50 experiments for testing. Data from each experiment was preprocessed by means of the DWT. The family

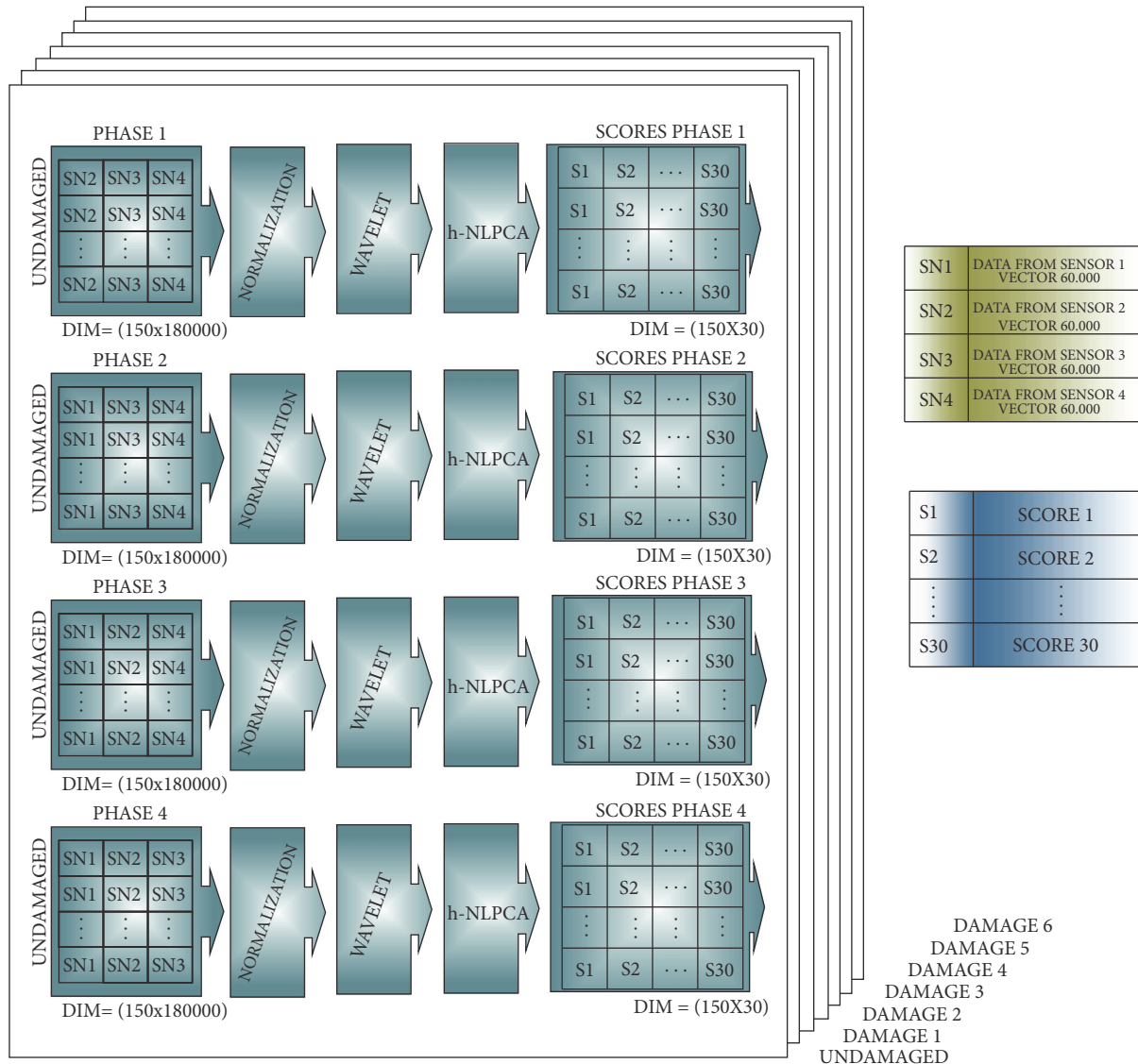


FIGURE 6: Data organization: h-NLPCA scores, before building the training vector.

Daubechies (db8) was chosen to obtain the approximation coefficients [46, 47]. This selection was applied since previous works demonstrated that this family contains most relevant information for this kind of applications. Coefficients are used to build the hierarchical nonlinear PCA model for each actuation phase. The architecture of the h-NLPCA is a five-layer nonlinear autoencoder network with 3-4-2-4-3 components as in [48]. As a result, three components by each actuation phase are used to build the feature vector that is used as the input for the training process to different machines. For the training part, the MATLAB classification learner app was used.

Subsequently, testing is performed by using data from the structure in an unknown structural state and projecting the information to the nonlinear components. The projected information, called scores, is used as the input to the trained machine to predict the kind of structural state. This procedure allows to detect and classify the structural state.

Several machines were trained to determine the elements in the feature vector, i.e., to determine the influence and the number of scores to use by actuation phase and the number of experiments for an adequate training machine. Table 2 shows the results in the prediction process when two scores by each actuation phase and fifty experiments are used in the training step. During the prediction, one hundred experiments per damage are used. Twenty supervised learning machines were training using MATLAB's classification learner toolbox.

As it is possible to observe, all structural states are not properly predicted in all the trained machines, this means that a low number of scores affect the classification process. Table 3 shows the results when the number of scores per actuation phase are increased to five. As it is possible to observe, prediction improves in most of the machines, however, it is necessary to determine an adequate number of scores, because when it is increased could produce machine overfitting, and the learning may be poor. This is that the

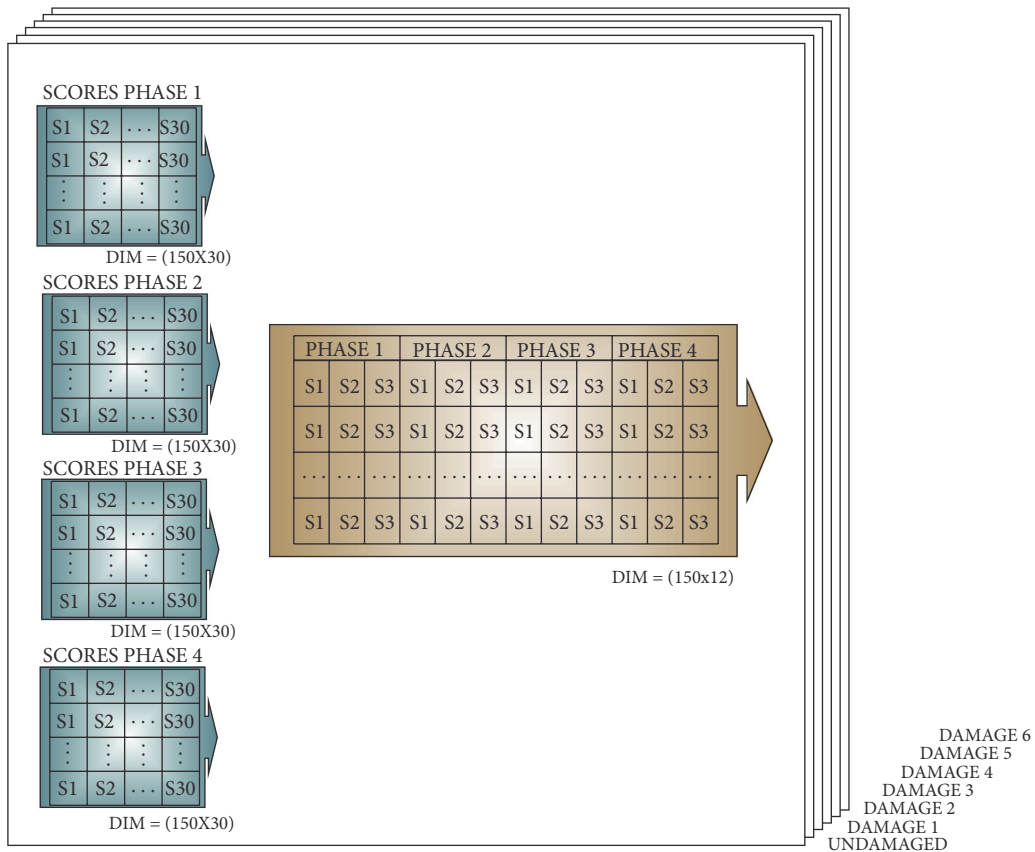


FIGURE 7: Assembled training vector.

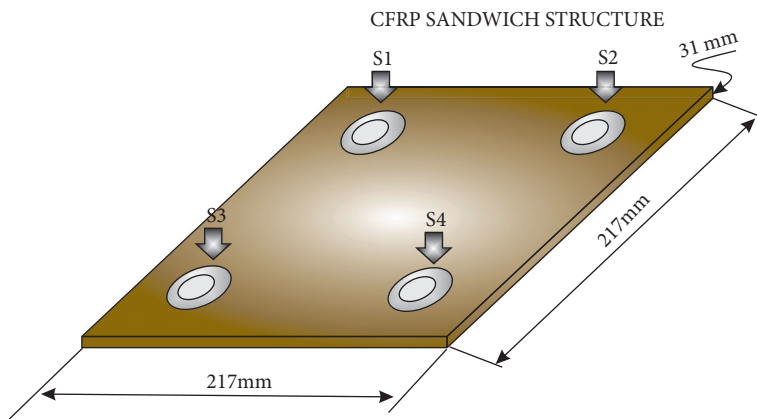


FIGURE 8: Specimen I: CFRP sandwich structure and PZT distribution.

mistakes are added to others predictions and growing up the uncertainty.

Consistent with previous research, fine k NN and weighted k NN showed better results in the classification. However, when the number of scores is increased, other machines such as bagged trees and subspace k NN significantly improved their performance. Following this analysis, three scores were defined as the number of scores to use because present similar results to the obtained with a greater number of scores.

Some consideration about the algorithms can be summarized as follows, the k nearest neighbour (k NN) classifier is an algorithm recommended to work with low dimensional data. Particularly, in this kind of machine, the number of neighbours have an effect over the response so, in general, the use of a reduced number of neighbours improve the outcome. Decision trees (DT) are a different kind of machine. In this case, DT is a classification mechanism that allows us to construct a predictive model where the value of splits can increase or decrease the flexibility of this algorithm, as well

TABLE 2: Behavior of machines with two scores per sensor (specimen 1, four sensors).

Machine type	UND	DMG1	DMG2	DMG3	DMG4	DMG5	DMG6
Complex Tree	90%	99%	13%	92%	100%	90%	100%
Medium Tree	90%	88%	13%	92%	100%	90%	100%
Simple Tree	90%	99%	0%	0%	100%	90%	100%
Linear SVM	96%	98%	81%	95%	99%	99%	100%
Quadratic SVM	96%	98%	96%	95%	99%	99%	100%
Cubic SVM	96%	99%	98%	95%	99%	99%	100%
Fine Gaussian SVM	68%	100%	57%	87%	79%	78%	99%
Medium Gaussian SVM	97%	100%	76%	100%	97%	98%	100%
Coarse Gaussian SVM	95%	98%	94%	96%	99%	99%	100%
Fine KNN	97%	100%	96%	98%	99%	100%	100%
Medium KNN	95%	100%	93%	94%	99%	100%	100%
Coarse KNN	91%	100%	85%	80%	99%	100%	94%
Cosine KNN	95%	100%	74%	89%	99%	100%	100%
Cubic KNN	95%	99%	89%	93%	99%	99%	100%
Weighted KNN	95%	100%	95%	97%	99%	100%	100%
Boosted Trees	90%	100%	20%	1%	100%	98%	100%
Bagged Trees	99%	100%	71%	95%	100%	100%	100%
Subspace Discriminant	97%	100%	64%	97%	100%	100%	100%
Subspace KNN	97%	100%	82%	98%	100%	100%	100%
Rusboosted Trees	90%	100%	0%	0%	0%	0%	0%

TABLE 3: Behavior of machines with five scores per sensor (specimen 1, four sensors).

Machine type	UND	DMG1	DMG2	DMG3	DMG4	DMG5	DMG6
Complex Tree	90%	99%	18%	99%	99%	97%	100%
Medium Tree	90%	99%	18%	99%	99%	97%	100%
Simple Tree	90%	99%	0%	100%	0%	97%	100%
Linear SVM	97%	100%	100%	99%	99%	99%	100%
Quadratic SVM	97%	100%	100%	99%	99%	99%	100%
Cubic SVM	97%	100%	100%	99%	99%	99%	100%
Fine Gaussian SVM	100%	9%	8%	28%	8%	30%	56%
Medium Gaussian SVM	99%	100%	98%	99%	99%	98%	100%
Coarse Gaussian SVM	98%	100%	100%	100%	99%	100%	100%
Fine KNN	97%	100%	100%	100%	99%	100%	100%
Medium KNN	97%	100%	100%	100%	99%	100%	100%
Coarse KNN	93%	100%	100%	99%	97%	100%	100%
Cosine KNN	96%	100%	100%	100%	99%	100%	100%
Cubic KNN	95%	100%	100%	100%	99%	99%	100%
Weighted KNN	97%	100%	100%	100%	99%	100%	100%
Boosted Trees	90%	100%	0%	100%	0%	100%	100%
Bagged Trees	99%	100%	100%	100%	100%	100%	100%
Subspace Discriminant	98%	100%	100%	100%	99%	100%	100%
Subspace KNN	98%	100%	100%	100%	99%	100%	100%
Rusboosted Trees	90%	100%	0%	0%	0%	0%	0%

as the use of various trees (ensemble). Other kind of machine explored in this paper is the RUS (Random Under Sampling) algorithm in RUSBoost, which is a mechanism to eliminate data distribution imbalances.

Figures 10 and 11 show the results in the damage classification process for fine k NN, weighted k NN, simple tree, and rusboosted trees. Detailed information about the definition of these machines can be found in [7, 44, 49]. As it is

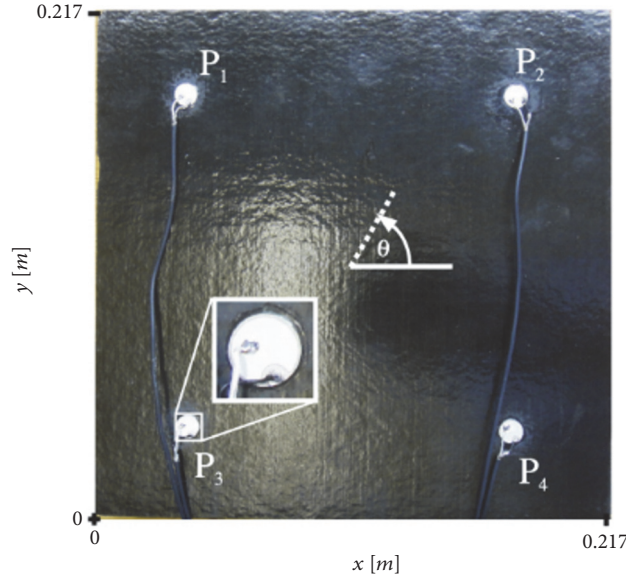


FIGURE 9: CRFP sandwich structure.

		Confusion Matrix - Fine KNN							Confusion Matrix - Weighted KNN															
True class ->	UND	98.00% (49)	2.00% (1)	0	0	0	0	0	98.00% (49)	2.00% (1)	0	0	0	0	0	96.00% (48)	4.00% (2)	0	0	0	0	0	0	
	DMG1	0	100.00% (50)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	
	DMG2	0	0	98.00% (49)	2.00% (1)	0	0	0	98.00% (49)	2.00% (1)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	0
	DMG3	0	0	0	100.00% (50)	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	0
	DMG4	0	0	0	0	100.00% (50)	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	0
	DMG5	0	0	0	0	0	100.00% (50)	0	100.00% (50)	0.00% (0)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	0
	DMG6	0	0	0	0	0	0	100.00% (50)	100.00% (50)	0.00% (0)	0	0	0	0	0	100.00% (50)	0.00% (0)	0	0	0	0	0	0	0
		TPR/FNR							TPR/FNR															
Predicted class ->		UND	DMG1	DMG2	DMG3	DMG4	DMG5	DMG6	UND	DMG1	DMG2	DMG3	DMG4	DMG5	DMG6	UND	DMG1	DMG2	DMG3	DMG4	DMG5	DMG6	TPR/FNR	

FIGURE 10: Confusion matrices for fine kNN (left) and weighted kNN (right) machines.

possible to observe in Figure 10, both fine kNN and weighted kNN presented some of the best results since in most of the experiments, the classification was properly performed, verifying its good behaviour like a statistical classifier [50]. For instance, in the fine kNN classifier, 348 cases have been correctly classified out of 350 cases. This magnitude represents 99,4% of correct decisions. It is worth noting that the specimen with damage is never confused with the healthy state of the structure thus leading to an absence of missing faults. The only misclassification between damages occurs with a sample corresponding to damage 2 that is classified as damage 3. Similar results are obtained when the weighted kNN is considered as the classifier. In this case, 348 cases have been correctly classified out of 350 cases, which represents 99,4% of correct decisions, too. However, in this case, all the damages are perfectly classified. The number of false alarms

is quite reduced in both cases: 1 out of 50 (2%) and 2 out of 50 (4%), with respect to fine kNN and weighted kNN, respectively.

Worst results in the classification are obtained when rusboosted trees and simple tree machines are used. These results are summarized through the corresponding confusion matrices in Figure 11. The overall accuracy is 26,3% and 88,9%, in the case of rusboosted trees and simple tree machines, respectively. The classification is especially unacceptable in the case of rusboosted trees where damages 2 to 6 are all misclassified.

4.3. *Specimen 2: CFRP Composite Plate.* The second structure, shown in Figures 12 and 13, corresponds to a CFRP plate made of 4 equal layers and stacking of $[0^\circ 90^\circ 90^\circ 0^\circ]$. Dimensions are 200 mm x 250 mm with a thickness of 1.7 mm

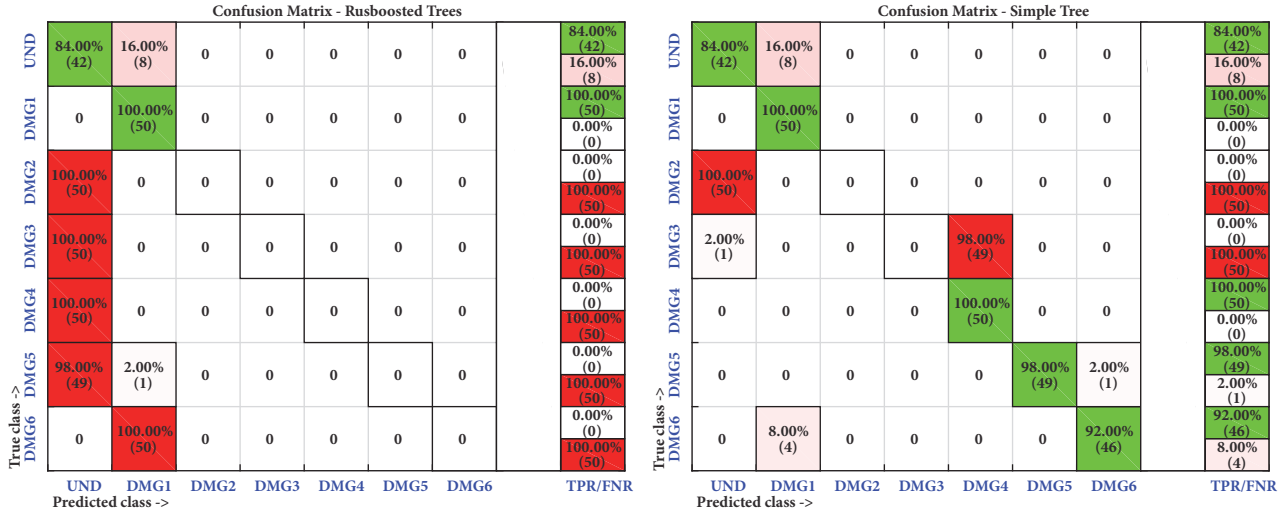


FIGURE 11: Confusion matrices for rusboosted trees (left) and simple tree (right) machines.

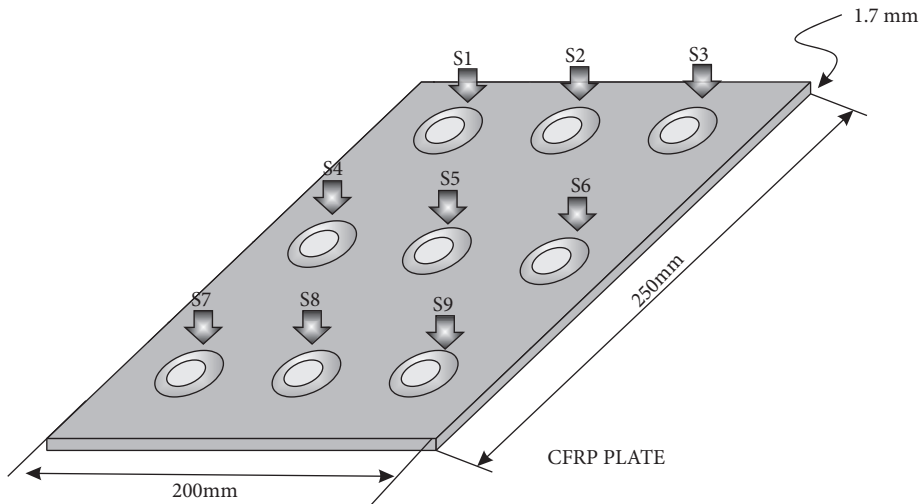


FIGURE 12: Specimen 2: CFRP composite plate, sensors distribution.

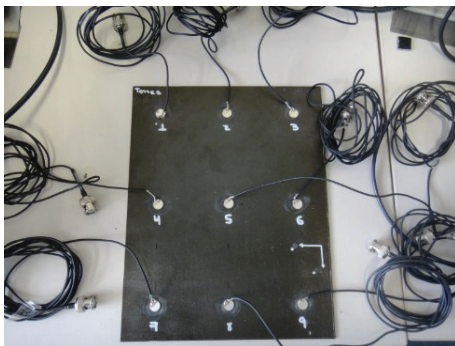


FIGURE 13: CFRP composite plate.

and a density of 1.700 kg/m^3 . Nominal material parameters of the unidirectional (UD) layers are $E1 = 122 \text{ GPa}$, $E2 = 10 \text{ GPa}$, $\nu12 = 0,33$, $\nu13 = 0,3$, $\nu23 = 0,34$, $G12 = G13 = 7,4 \text{ GPa}$, and $G23 = 5,4 \text{ GPa}$.

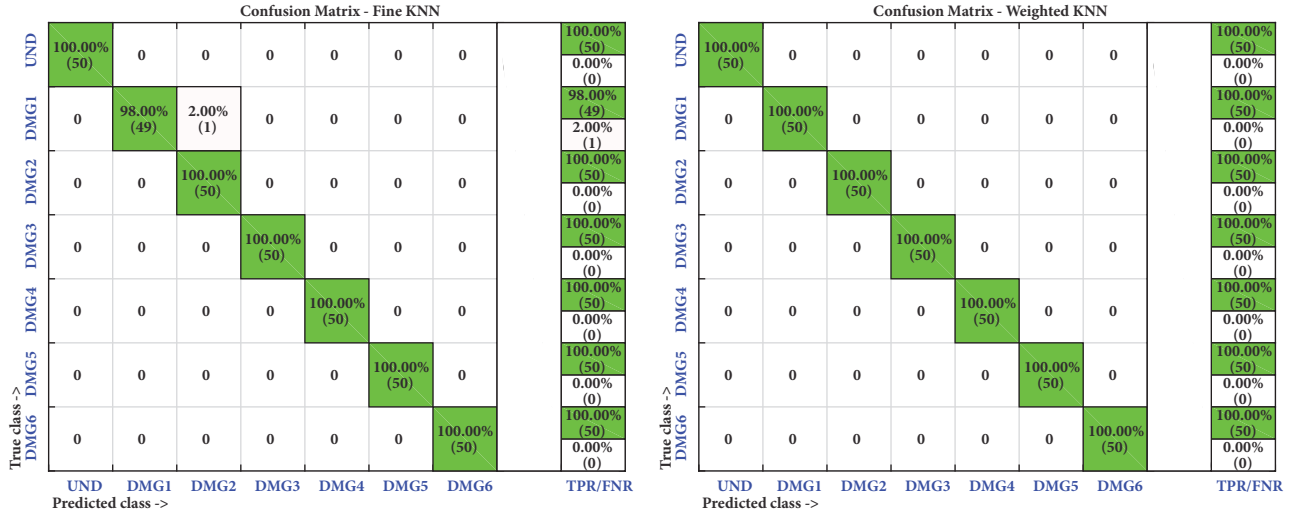
This structure was instrumented with nine piezoelectric transducers PIC-151 from PI Ceramics which are attached to the surface of the structure as it is shown in Figure 12. Damage on the tested composite was simulated by localizing masses at different positions as described in Table 4.

The excitation signal is a 12 V Hamming windowed cosine train signal with 5 cycles, 150 experiments have been performed and signals from sensors have been also recorded per sensor-actuator configuration to each structural state. To determine the carrier central frequency for the actuation signal in each structure, a frequency sweep was performed and spectral analysis of each signal was analysed in order to determine the optimal excitability frequency (structure and sensors) where the obtained signals have a signal/noise ratio that helps to the data analysis. The carrier frequency in this specimen was found to be 30 kHz. A photo of this second specimen can be found in Figure 13.

As with the previous specimen, several machines were trained and three scores were used per actuation phase.

TABLE 4: Damages in the CFRP composite plate.

Damage number	Damage location between sensors	X position [mm]	Y position [mm]
1	Sensors 1-2	65	220
2	Sensors 2-3	135	220
3	Sensors 3-6	170	172.5
4	Sensors 6-9	170	66.5
5	Sensors 5-8	100	66.5
6	Sensors 5-4	65	125

FIGURE 14: Confusion matrices for fine k NN (left) and weighted k NN (right) machines.

For this second experiment and for the case of fine k NN and weighted k NN, the results are even better (Figure 14). More precisely, in the fine k NN classifier, 349 cases have been correctly classified out of 350 cases. This magnitude represents 99,7% of correct decisions. It is worth noting that the specimen with damage is never confused with the healthy state of the structure thus leading to an absence of missing faults. The only misclassification between damages occurs with a sample corresponding to damage 1 that is classified as damage 2. A perfect classification is obtained when the weighted k NN is considered as the classifier. In this case, 350 cases have been correctly classified out of 350 cases, which represents 100% of correct decisions. With respect to this second specimen, false alarms are no longer present.

Worst results in the classification are obtained when rusboosted trees and simple tree machines are used. These results are summarized through the corresponding confusion matrices in Figure 15. The overall accuracy is 28,3% and 70,9%, in the case of rusboosted trees and simple tree machines, respectively. The classification is especially unacceptable in the case of rusboosted trees where damages 2 to 6 are all misclassified. Although the percentage of correct decisions fluctuates between 28,3% and 70,9%, both machines are able to accurately identify the structure with no damage.

In general, the behaviour of these four machines in this paper with respect to both specimens is coherent with previous results in the literature. For instance, in the work

of Vitola et al. [44, 49], a distributed sensor network is used to detect and classify structural changes with and without the influence of environmental conditions. Although in those papers how the data is collected and preprocessed differs significantly from the current work, the performance of both fine k NN and weighted k NN is similar.

5. Conclusions

In this work, a damage classification methodology has been introduced. The proposed methodology includes the use of a piezoelectric sensor network, discrete wavelet transform, hierarchical nonlinear PCA, and machine learning approaches. The methodology has been validated with excellent results showing its capability for damage classification tasks. Although different machines were trained, only the best two and the worst two of them were included in the paper, showing that the best results were obtained with fine k NN and weighted k NN machines and worst results are obtained by the use of trees. This is because the way as the data is organized by the different machines as was introduced along the paper.

In order to work with machine learning algorithms, it is very important to select the training data in a proper way. Otherwise, results in the trained machine can be different to the system expectations. The nonlinear scores demonstrated that the extracted information was very useful, since these scores reduced significantly the information by facilitating

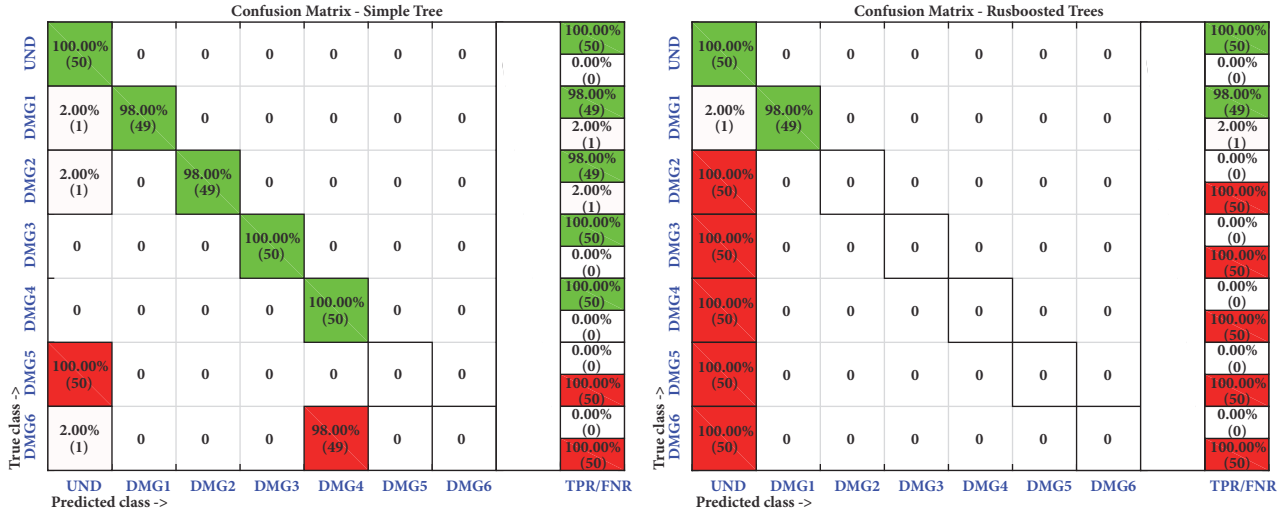


FIGURE 15: Confusion matrices for simple tree (left) and rusboosted trees (right) machines.

the training and reducing the error possibilities in the predictions. In all studied cases, the use of three nonlinear scores demonstrates to be enough for building the featured vector by fusing data from all actuation phases.

The k nearest neighbours algorithm has also shown to be an efficient and useful mechanism to applications in structural health monitoring. Results in this work also indicated its usefulness with the use of nonlinear features.

The use of neural networks in this work is only considered to obtain the nonlinear components; however it is expected to work in a near future with different neural networks approaches for classification.

Although this work is not focused on the study of the relationship between the number and the location of sensors but on the damage classification methodology, the inspected structures allow us to extend the idea of the usefulness of this methodology by the following differences between the validations:

- (1) Different structures with different materials and configurations.
- (2) Structures with real (delamination and cracks) and simulated damage (added mass in different locations).
- (3) Progressive damage.
- (4) Structures with different sizes.
- (5) Structures inspected with a different number of sensors at different locations. In the case of the first structure only four sensors were used and the second structure was inspected with nine sensors.

Future work will involve the influence of the number of sensors and location. However this study allows us to observe that the methodology can be used with similar results in different structures with different number and position of the sensors and with different kind of damages because of the pattern recognition approach.

Abbreviations

- CFRP: Carbon fibre-reinforced plastic
- db8: Daubechies
- DWT: Discrete wavelet transform
- h-NLPCA: Hierarchical nonlinear principal component analysis
- k NN: k -Nearest neighbours
- ML: Machine learning
- MLP: Multilayered perceptron
- mm: Millimetres
- MSE: Mean square error
- PCA: Principal component analysis
- PZT: Piezoelectric sensor
- SHM: Structural health monitoring.

Data Availability

Most of the data used in this paper was collected in several research stays and we are authorized to use the data and develop our methodologies but not to share the original data publicly.

Disclosure

The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results. A preliminary version of the manuscript was present in the 11th International Workshop on Structural Health Monitoring organized by the Stanford University in the year 2017 [47].

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science, Innovation, and Universities, the Agencia Estatal de Investigación (AEI), and the European Regional Development Fund (ERDF/FEDER) through the research Projects DPI2014-58427-C2-1-R and DPI2017-82930-C2-1-R. This work also is supported by Universidad Santo Tomás through Grant FODEIN 2018, Project Code FODEIN 1854504.

References

- [1] K. Van Buren, J. Reilly, K. Neal, H. Edwards, and F. Hemez, "Guaranteeing robustness of structural condition monitoring to environmental variability," *Journal of Sound and Vibration*, vol. 386, pp. 134–148, 2017.
- [2] K. Worden and C. R. Farrar, *Structural Health Monitoring: A Machine Learning Perspective*, 2013.
- [3] D. Balageas, C. P. Fritzen, and A. Güemes, *Structural Health Monitoring*, Hermes Science Publishing, 2006.
- [4] M. Yarnold and F. Moon, "Temperature-based structural health monitoring baseline for long-span bridges," *Engineering Structures*, vol. 86, pp. 157–167, 2015.
- [5] M. M. Alamdari, T. Rakotoarivelo, and N. L. D. Khoa, "A spectral-based clustering for structural health monitoring of the Sydney Harbour Bridge," *Mechanical Systems and Signal Processing*, vol. 87, pp. 384–400, 2017.
- [6] J. J. McCullagh, T. Galchev, R. L. Peterson et al., "Long-term testing of a vibration harvesting system for the structural health monitoring of bridges," *Sensors and Actuators A: Physical*, vol. 217, pp. 139–150, 2014.
- [7] D. A. Tibaduiza, *Design and Validation of a Structural Health Monitoring System for Aeronautical Structures*, Universitat Politècnica de Catalunya, 2013.
- [8] R. K. Neerukatti, K. C. Liu, N. Kovvali, and A. Chattopadhyay, "Fatigue life prediction using hybrid prognosis for structural health monitoring," *Journal of Aerospace Information Systems*, vol. 11, no. 4, pp. 211–231, 2014.
- [9] V. Giurgiutiu, *Structural Health Monitoring of Aerospace Composites*, Academic Press - ELSEVIER, 2015.
- [10] A. Korobenko, M. Pigazzini, V. Singh et al., "Dynamic-Data-Driven Damage Prediction in Aerospace Composite Structures," in *Proceedings of the 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2016.
- [11] I. Antoniadou, N. Dervilis, E. Papatheou, A. E. Maguire, and K. Worden, "Aspects of structural health and condition monitoring of offshore wind turbines," *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences*, 2015.
- [12] S. Bogoevska, M. Spiridonakos, E. Chatzi, E. Dumova-Jovanoska, and R. Höffer, "A data-driven diagnostic framework for wind turbine structures: A holistic approach," *Sensors*, vol. 17, no. 4, 2017.
- [13] F. Pozo and Y. Vidal, "Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing," *Energies*, vol. 9, no. 1, p. 3, 2016.
- [14] M. Anaya, D. A. Tibaduiza, and F. Pozo, "Detection and classification of structural changes using artificial immune systems and fuzzy clustering," *International Journal of Bio-Inspired Computation*, vol. 9, no. 1, pp. 35–52, 2017.
- [15] M. A. Torres-Arredondo, *Acoustic Emission Testing and Acousto-Ultrasonics for Structural Health Monitoring*, University of Siegen, Siegen, Germany, 2013.
- [16] M. J. Shensa, "The discrete wavelet transform: wedding the Á trous and Mallat algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2464–2482, 1992.
- [17] A. Graps, "An introduction to wavelets," *IEEE Computational Science & Engineering*, vol. 2, no. 2, pp. 50–61, 1995.
- [18] H. Z. Hosseinabadi, B. Nazari, R. Amirfattahi, H. R. Mirdamadi, and A. R. Sadri, "Wavelet network approach for structural damage identification using guided ultrasonic waves," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 7, pp. 1680–1692, 2014.
- [19] X. Chen, X. Li, S. Wang, Z. Yang, B. Chen, and Z. He, "Composite damage detection based on redundant second-generation wavelet transform and fractal dimension tomography algorithm of lamb wave," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 5, pp. 1354–1363, 2013.
- [20] A. Medda, E. Chicken, and V. DeBrunner, "Sigma-Sampling Wavelet Denoising for Structural Health Monitoring," in *Proceedings of the 2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, pp. 119–122, Madison, WI, USA, August 2007.
- [21] M. Golub, A. Shpak, I. Buethe, C. Fritzen, H. Jung, and J. Moll, "Continuous wavelet transform application in diagnostics of piezoelectric wafer active sensors," *International Conference Days on Diffraction*, vol. 2013, pp. 59–64, 2013.
- [22] H. Jeong, "Analysis of plate wave propagation in anisotropic laminates using a wavelet transform," *NDT & E International*, vol. 34, no. 3, pp. 185–190, 2001.
- [23] M. A. Torres-Arredondo, I. Buethe, D. A. Tibaduiza, J. Rodellar, and C.-P. Fritzen, "Damage detection and classification in pipework using acousto-ultrasonics and non-linear data-driven modelling," *Journal of Civil Structural Health Monitoring*, vol. 3, no. 4, pp. 297–306, 2013.
- [24] M. Scholz, F. Kaplan, C. L. Guy, J. Kopka, and J. Selbig, "Non-linear PCA: A missing data approach," *Bioinformatics*, vol. 21, no. 20, pp. 3887–3895, 2005.
- [25] A. D. Santos, M. F. Silva, C. S. Sales, J. C. Costa, and E. Figueiredo, "Applicability of linear and nonlinear principal component analysis for damage detection," in *Proceedings of the 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 869–874, Pisa, Italy, May 2015.
- [26] M. Scholz, *Approaches to Analyse and Interpret Biological Profile Data*, Max Planck Institute of Molecular Plant Physiology, Potsdam University, 2006.
- [27] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [28] M. A. Torres, D. A. Tibaduiza, L. E. Mujica, J. Rodellar, and C. P. Fritzen, "Damage Assessment in a Stiffened Composite Panel using Non-Linear Data-Driven Modelling and Ultrasonic Guided Waves," in *Proceedings of the 4th Int. Symp. NDT Aerospace*, Ausburg, Germany, 2012.
- [29] Z. Liu and N. Mrad, "Validation of strain gauges for structural health monitoring with bayesian belief networks," *IEEE Sensors Journal*, vol. 13, no. 1, pp. 400–407, 2013.
- [30] H. HoThu and A. Mita, "Damage Detection Method Using Support Vector Machine and First Three Natural Frequencies

- for Shear Structures,” *Open Journal of Civil Engineering*, vol. 03, no. 02, pp. 104–112, 2013.
- [31] J. Zhang and S. Zhou, *Structure Health Monitoring in Extreme Events From Machine Learning Perspective*.
- [32] J. Zhang and Z. Hou, “Application of Artificial Immune System in Structural Health Monitoring,” *Journal of Structures*, vol. 2014, Article ID 709127, 14 pages, 2014.
- [33] K. Murphy, *Machine Learning: A Probabilistic Perspective*, 1991.
- [34] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 4, Springer, New York, NY, USA, 2006.
- [35] W. Nick, J. Shelton, K. Asamene, and A. C. Esterline, *A Study of Supervised Machine Learning Techniques for Structural Health Monitoring*, 2015.
- [36] D. Charles, C. Fyfe, D. Livingstone, and S. McGlinchey, *Biologically Inspired Artificial Intelligence for Computer Games*, IGI Global, 2008.
- [37] W. Nick, K. Asamene, G. Bullock, A. Esterline, and M. Sundaresan, “A Study of Machine Learning Techniques for Detecting and Classifying Structural Damage,” *International Journal of Machine Learning and Computing*, vol. 5, no. 4, pp. 313–318, 2015.
- [38] B. Clarke, E. Fokoué, and H. H. Zhang, *Principles and Theory for Data Mining and Machine Learning*, Springer Science & Business Media, 2009.
- [39] K. R. Mulligan, C. Yang, N. Quaegebeur, and P. Masson, “A data-driven method for predicting structural degradation using a piezoceramic array,” *International Journal of Prognostics and Health Management*, vol. 4, pp. 1–14, 2013.
- [40] F. G. Baptista and J. V. Filho, “Transducer loading effect on the performance of PZT-based SHM systems,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 4, pp. 933–941, 2010.
- [41] F. G. Baptista and J. V. Filho, “Optimal frequency range selection for PZT transducers in impedance-based SHM systems,” *Sensors Journal*, vol. 10, pp. 1297–1303, 2010.
- [42] M.-A. Torres-Arredondo, D.-A. Tibaduiza, M. McGugan et al., “Multivariate data-driven modelling and pattern recognition for damage detection and identification for acoustic emission and acousto-ultrasonics,” *Smart Materials and Structures*, vol. 22, no. 10, 2013.
- [43] L. E. Mujica, D. A. Tibaduiza, and J. Rodellar, “Data-Driven Multiactuator Piezoelectric System for Structural Damage Localization,” in *Proceedings of the Fifth World Conf. Struct. Control Monit*, Tokio, Japan, 2010.
- [44] J. Vitola, F. Pozo, D. Tibaduiza, and M. Anaya, “A Sensor Data Fusion System Based on k-Nearest Neighbor Pattern Classification for Structural Health Monitoring Applications,” *Sensors*, vol. 17, no. 2, pp. 1–26, 2017.
- [45] M. Anaya, D. Tibaduiza, and F. Pozo, “A Bioinspired Methodology Based on an Artificial Immune System for Damage Detection in Structural Health Monitoring,” *Shock and Vibration*, vol. 2015, Article ID 648097, 15 pages, 2015.
- [46] M. A. Torres-Arredondo and D. A. Tibaduiza-Burgos, “An acousto-ultrasonics approach for probabilistic modelling and inference based on Gaussian processes,” *Structural Control and Health Monitoring*, vol. 25, no. 6, Article ID e2178, 2018.
- [47] D. A. Tibaduiza, M. A. Torres, J. Vitola, M. Anaya, and F. Pozo, “Non-linear damage classification based on machine learning and damage indices,” in *Proceedings of the 11th International Workshop on Structural Health Monitoring 2017: Real-Time Material State Awareness and Data-Driven Safety Assurance, IWSHM 2017*, pp. 2096–2102, Stanford, Calif, USA, September 2017.
- [48] M. Scholz, R. Vigário, and R. Vigário, “Non-linear PCA: a new hierarchical approach,” in *Proceedings of the 10th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 2002.
- [49] J. Vitola, F. Pozo, D. Tibaduiza, and M. Anaya, “Distributed piezoelectric sensor system for damage identification in structures subjected to temperature changes,” *Sensors*, vol. 17, no. 6, pp. 1–26, 2017.
- [50] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, “Neural and statistical classifiers - Taxonomy and two case studies,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 1, pp. 5–17, 1997.

Research Article

Artificial Neural Networks Based Friction Law for Elastomeric Materials Applied in Finite Element Sliding Contact Simulations

Aleksandra Serafińska, Wolfgang Graf, and Michael Kaliske 

Institute for Structural Analysis, Technische Universität Dresden, D-01062 Dresden, Germany

Correspondence should be addressed to Michael Kaliske; michael.kaliske@tu-dresden.de

Received 5 July 2018; Accepted 2 October 2018; Published 19 November 2018

Academic Editor: Francisco J. Montáns

Copyright © 2018 Aleksandra Serafińska et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A realistic characterization of the frictional behaviour of materials and mechanical systems is of prime importance for the assessment of their contact interaction properties, especially in the context of undesired temperature rise or intensive wear leading to service life reduction. A characteristic tribological property of elastomeric materials is the dependency of the friction coefficient on the local contact pressure, sliding velocity, and temperature in the contact interface. Thus, the friction coefficient is not constant in the entire contact area but varies according to the magnitudes of the aforementioned three influencing factors. In this contribution, a friction law based on artificial neural networks (ANN) is presented, which is able to capture the nonlinear dependencies of the friction coefficient on the contact pressure, sliding velocity, and temperature. Due to an extraordinary adaptivity of the ANN structure, these nonlinear relations stemming from experimental data can be modelled properly within the introduced friction law, in contrast to other friction formulations, which are limited by the fitting quality of their parameters. The ANN based friction law is implemented into a contact formulation of the finite element method (FEM). Especially, the linearization of contact contributions to the weak form of momentum balance equation, required for the FEM, is developed taking into account the differentiability of the ANN. The applicability of the developed friction law within the finite element analysis of tires as well as within sliding simulations of rubber elements is presented in this paper.

1. Introduction

In this contribution, a novel method enabling to capture the thermo-mechanical frictional characteristics of materials and mechanical systems is developed. An accurate modelling of frictional properties is crucial for the assessment of structural durability, since the friction energy dissipation in the contact interface can lead to pronounced wear [1] and temperature increase. Thereby, the intensification of both these effects results commonly in a material failure or degradation of surfaces in contact.

In order to model the friction phenomena of elastomeric structures in its whole complexity, the dependency of the friction coefficient on the contact pressure, sliding velocity, and temperature as well as on the topology of contacting surfaces has to be captured.

When considering the friction of elastomeric structures with respect to surface roughness, two contributions

determining the magnitude of the friction coefficient can be distinguished, the hysteresis and adhesion contribution [2]. Hysteresis friction is related to an internal energy dissipation appearing in viscoelastic materials during sliding over rough surfaces. On the other hand, adhesion friction results from intermolecular, distance-dependent interaction forces between two surfaces, which are pronounced especially on flat surfaces.

Analytical approaches introducing friction theories considering surface roughness are proposed by Klüppel and Heinrich [2, 3] as well as by Persson [4]. Furthermore, numerical approaches for the modelling of friction are developed, which are based on the finite element method, multiscale analysis, and homogenization; see [5–7].

For the analysis of the frictional behaviour of mechanical systems, which are commonly performed at a structural level, macroscopic friction laws are predominantly applied. These formulations do not consider surface roughness and

frictional properties at multiple scales. They rather provide formulations for the dependency of tangential stresses or the friction coefficient on the local contact pressure, sliding velocity, and temperature in the contact interface. These dependencies are developed for certain materials in contact. Such macroscopic friction laws are proposed in [8–10].

The friction law introduced in this contribution also belongs to the category of macroscopic friction laws. Though, while other macroscopic equations for friction commonly take into account only one or two mentioned factors, which influence the magnitude of the friction coefficient, the herein presented method enables to account for all three factors and can be even extended to more aspects, if required.

In the proposed constitutive equation for friction, meta-models in form of response surfaces based on artificial neural networks [11] are implemented within the friction formulation. Due to an outstanding adaptivity of the ANNs, the fitting quality to the experimental results is commonly higher than the one obtained for other friction formulations. Furthermore, the differentiability of the ANN formulation enables a direct specification of linearization terms of contact contributions to the weak form of momentum balance equation, required for the Newton method to find the solution. Thus, a straightforward implementation into the FEM is feasible.

A further advantage of the introduced friction law is its temperature dependency, which enables the application of the law within thermo-mechanically coupled FE analyses. This fact is of significant importance, since the properties of elastomeric materials change distinctively due to the influence of temperature.

The benefits of the utilization of ANNs in various engineering tasks and disciplines have been noticed in many publications; e.g., in [12], an overview of the application of ANNs in civil engineering is presented. ANNs have been introduced to material law formulations; see [13–15]. Furthermore, ANNs are commonly employed for the approximation of a response surface, which can substitute the computational model of the structural analysis, compare [16, 17]. Moreover, the application of ANNs to structural analysis with uncertain data [18] and to pattern recognition approaches (see [19]) is known from the literature, though, an ANN based friction law characterized by a consistent formulation for the FEM, applicable to thermo-mechanically coupled analysis, is still an unexplored topic and, thus, in the scope of this contribution.

This paper is structured as follows. In Section 2, the results of an experimental investigation of friction properties are presented and a friction coefficient response surface approximation on the basis of ANN is shown. In Section 3, the architecture and the training procedure of the applied feed-forward artificial neural network are introduced. Section 4 is devoted to the formulation of the proposed metamodel based friction law and its implementation into the framework of the FEM. Finally, in Section 5 three examples visualizing the results of the FE analysis of elastomeric structures under the application of the developed friction formulation are presented.

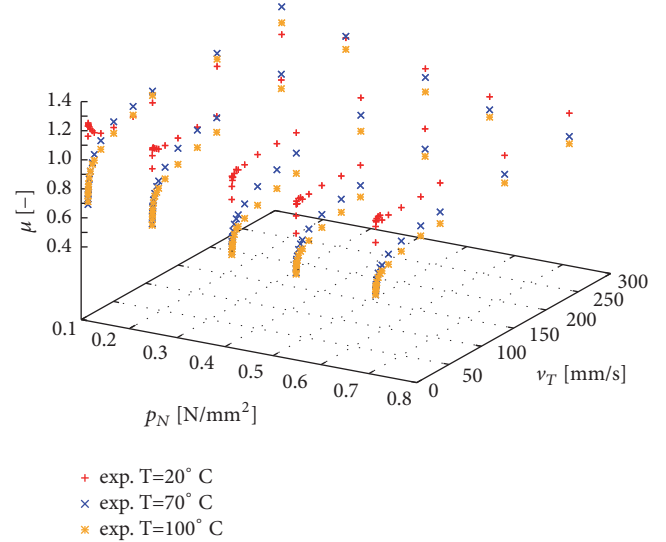


FIGURE 1: Friction coefficient at various contact pressures p_N , sliding velocities v_T , and temperatures T .

2. Friction Coefficient Response Surface Based on Experimental Data

The constitutive equation for friction investigated in this contribution is formulated as

$$\mathbf{t}_T = \mu(|p_N|, \|\mathbf{v}_T\|, T) |p_N| \frac{\mathbf{g}_T}{\|\mathbf{g}_T\|}, \quad (1)$$

where \mathbf{t}_T denotes the tangential stress, \mathbf{g}_T stands for the tangential slip, and μ is the friction coefficient. Thereby, the dependency of the friction coefficient on the normal pressure p_N , the sliding velocity v_T , and the temperature T is considered.

In order to identify the frictional properties of rubber and to account for the aforementioned friction coefficient formulation, experimental investigations including friction tests on a tribometer are conducted. Rubber samples of size 20x20x4 mm stemming from the tire tread pattern are subjected to frictional contact with a granite surface. Within the test, various temperature, velocity, and normal load (pressure) conditions are applied to identify the dependency of the friction coefficient on these factors. Especially, the temperature range $T = [20, 100]^\circ\text{C}$, the normal pressure range $p_N = [0.114, 0.714] \text{ N/mm}^2$, and the velocity range $v_T = [0.1, 300] \text{ mm/s}$ are tested. In each single experiment, the rubber block is subjected to a prescribed normal pressure, sliding velocity and temperature. Thereby, in every experiment nearly constant temperature conditions are provided since the rubber samples are preheated to a predefined temperature and the experiment is executed in a climate chamber with the same predefined indoor temperature. The tested rubber sample is fixed to a cantilever arm, which transmits the normal load and is subjected to contact with a disc rotating at a prescribed velocity.

In Figure 1, the results of the friction test are visualized. Thereby, a decrease of the friction coefficient with increasing

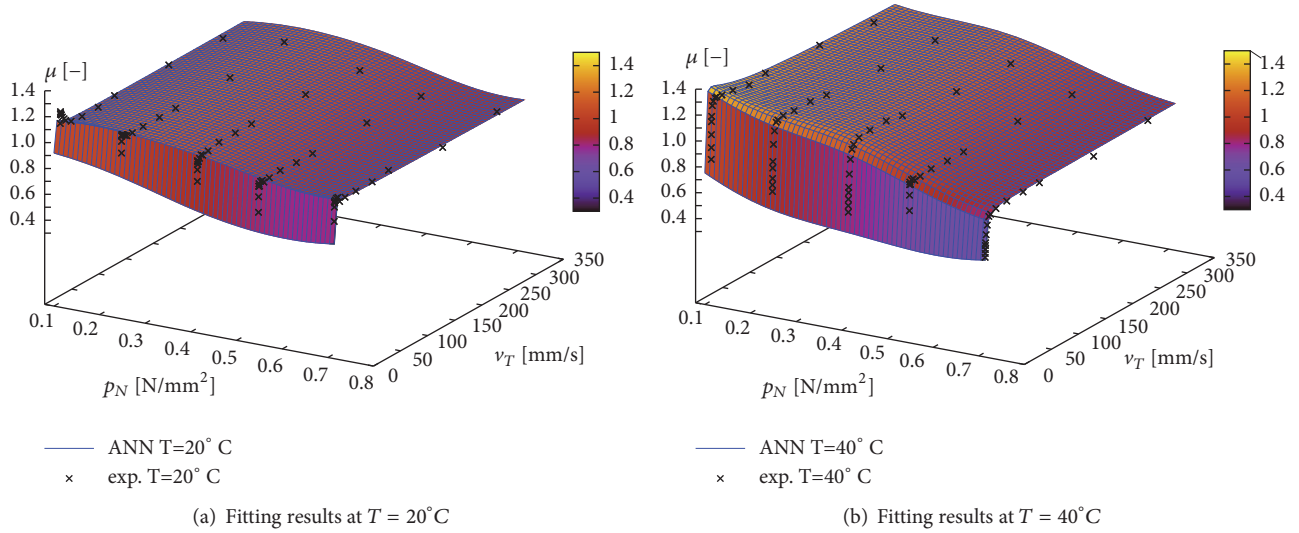


FIGURE 2: Friction coefficient response surfaces provided by ANN for experimental results at $T = 20^\circ\text{C}$ and $T = 40^\circ\text{C}$.

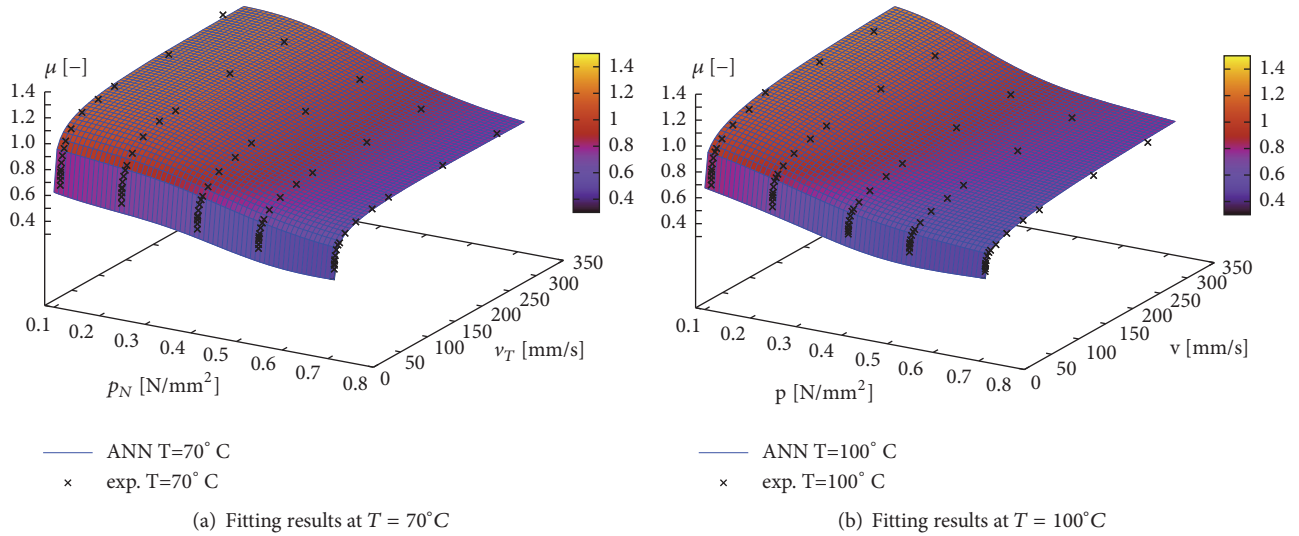


FIGURE 3: Friction coefficient response surfaces provided by ANN for experimental results at $T = 70^\circ\text{C}$ and $T = 100^\circ\text{C}$.

rubber temperature can be observed, except in the low pressure and high sliding velocity regimes. Furthermore, a steep ascent of the friction coefficient in the low sliding velocity range and a moderate ascending friction coefficient values in higher sliding velocity ranges are shown in Figure 1. Finally, the relation of the friction coefficient to the normal pressure is defined by decreasing μ magnitudes with the increasing p_N magnitudes.

In this contribution, a metamodel in form of an artificial neural network is developed, which is able to reproduce all the mentioned dependencies $\mu(p_N, v_T, T)$ with a very high accuracy. Furthermore, a single ANN is able to provide friction coefficient response surfaces at various temperatures, compare Figure 1. An advantage of such a response surface is the availability of the friction coefficient values interpolated between the measured friction coefficient magnitudes.

In Figures 2 and 3, the response surfaces provided by an ANN at tread block temperatures $T = 20, 40, 70, 100^\circ\text{C}$ are visualized. Due to its extraordinary adaptivity, a single ANN is able to reproduce various shapes of response surfaces at subsequent temperature levels.

Especially, the friction coefficient response surfaces at $T = 20^\circ\text{C}$ and $T = 40^\circ\text{C}$ are characterized by a steep ascent of the friction coefficient in low sliding velocity range and a flattening of the surface in higher sliding velocity range. On the other hand, the response surfaces at $T = 70^\circ\text{C}$ and $T = 100^\circ\text{C}$ show a smooth transition between the region with a steep ascent of the friction coefficient and the flatter region.

In order to visualize the high fitting quality of the ANN, cut-offs from response surfaces presented in Figures 2 and 3 at certain normal pressures $p_N = 0.248 \text{ N/mm}^2$ and $p_N = 0.548 \text{ N/mm}^2$ are prepared; see Figure 4. The friction

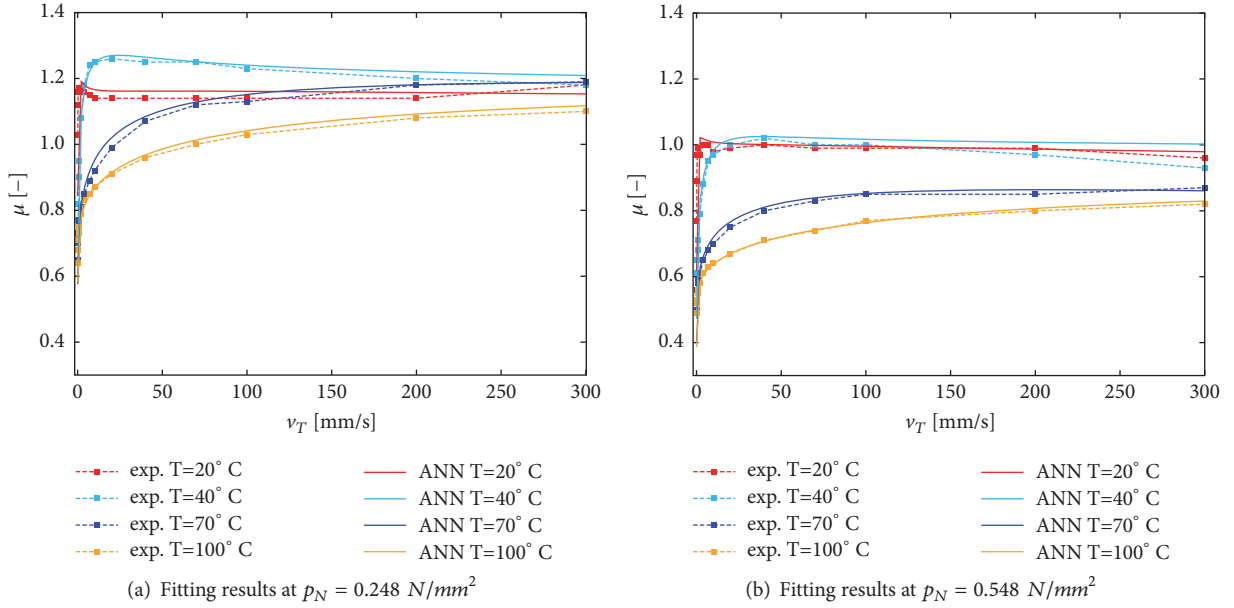


FIGURE 4: Friction coefficient μ provided by ANN versus experimental result at normal pressure $p_N = 0.248 \text{ N/mm}^2$ and $p_N = 0.548 \text{ N/mm}^2$ and at various temperatures.

coefficients computed by the ANN match the experimental results with a very high accuracy. All dependencies including the decrease of the friction coefficient with the increase of temperature and with the increase of normal pressure are captured by the developed metamodel.

As a reference solution for the developed metamodel based friction law, the Huemer friction law [8] enhanced by the temperature dependency is applied. The Huemer friction law is formulated as

$$\mathbf{t}_T = \mu_H |p_N| \frac{\Delta \mathbf{g}_T}{\|\Delta \mathbf{g}_T\|}, \quad (2)$$

where the friction coefficient μ_H is represented by

$$\mu_H = \frac{\alpha |p_N|^{n-1} + \beta}{a + b/\|\mathbf{v}_T\|^{1/m} + c/\|\mathbf{v}_T\|^{2/m}}. \quad (3)$$

The free parameters of the Huemer friction law $\alpha, \beta, a, b, c, m, n$ are fitted for the experimental data available at four temperature levels $T = 20, 40, 70, 100^\circ\text{C}$ corresponding to constant rubber block temperatures; see Figure 1. Taking into account the obtained four parameter sets, each parameter can be considered as temperature dependent $\alpha(T), \beta(T), a(T), b(T), c(T), m(T), n(T)$. In this way, the formulation of the Huemer friction law is extended by the temperature dependency.

In Figure 5, the friction coefficients obtained by the Huemer friction law are plotted versus the experimental results at normal pressures $p_N = 0.248 \text{ N/mm}^2$ and $p_N = 0.548 \text{ N/mm}^2$. Although for each temperature level, a separate set of parameters, which best approximates the experimental results, has been used, the fitting quality of the extended Huemer friction law is not as good as of the

metamodel based friction law. Especially, in the high sliding velocity regime, the friction coefficients computed within the Huemer friction law differ from the experimentally identified friction coefficients. Thereby, it should be noticed that a single metamodel is able to generate friction coefficient response surfaces at various temperature levels, whereas for the Huemer friction law, a separate parameter set is required to reproduce the friction coefficients at a certain temperature.

3. Architecture of Artificial Neural Networks

In this section, the architecture of the feedforward artificial neural network applied to the approximation of the friction coefficient response surfaces, presented in Section 2, is described. Since the structure of an ANN mimics the one of the human brain, two main structural components, neurons and synapses, can be distinguished, compare [11]. Thereby, the functionality of neurons as the information processing units involves a mapping of the neuron input signal x_i onto the output signal θ_j ; see Figure 6. Synapses, constituting the connections between two neurons i and j , are able to transmit the signal, which can be additionally strengthened or weakened according to the respective weight ω_{ij} of the synaptical connection.

Within a neuron j , the aggregation of the input signals x_i , which are multiplied by the synaptical weights ω_{ij} , is accomplished by the summing junction functionality

$$u_j = \sum_{i=1}^n \omega_{ij} x_i + v_j. \quad (4)$$

In (4), v_j stands for a bias term. Subsequently, the summing junction u_j is evaluated by an activation function \mathcal{F} yielding the neuron output signal θ_j ; see Figure 6. Within the

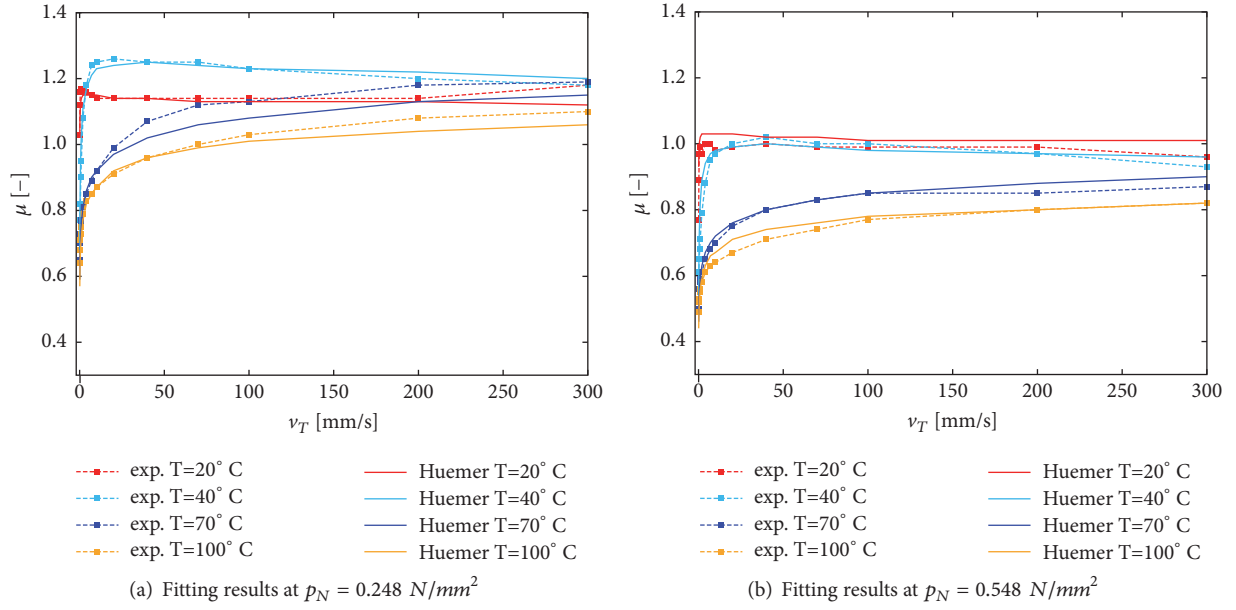


FIGURE 5: Friction coefficient μ provided by the Huemer friction law versus experimental result at normal pressure $p_N = 0.248 \text{ N/mm}^2$ and $p_N = 0.548 \text{ N/mm}^2$ and at various temperatures.

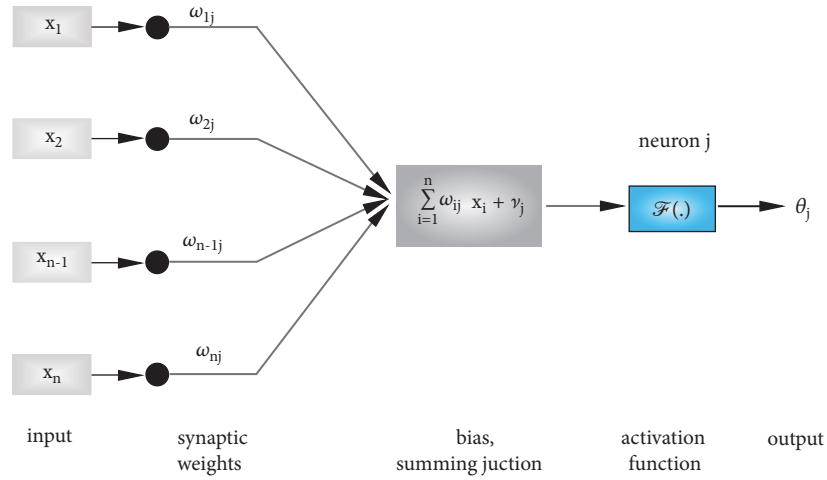


FIGURE 6: Information processing within a neuron.

developed ANN, logistic sigmoid functions are implemented as activation functions due to their advantageous properties, e.g., nonlinearity and differentiability

$$\theta_j = \mathcal{F}(u_j) = \frac{1}{1 + e^{-\sum_{i=1}^n \omega_{ij} x_i + v_j}}. \quad (5)$$

Within the artificial neural network architecture, neurons belong to one of three layer types, the input, the hidden and the output layer. For the approximation of the friction coefficient response surface an ANN with an input layer containing three neurons, which process the normal pressure p_N , the velocity v_T , and the temperature T as the input signals, is applied; see Figure 7. The selected ANN contains one hidden layer with 14 neurons and an output layer with a single

neuron returning the computed friction coefficient μ_{ann} as the output signal θ_k .

Since in this approach the feedforward neural networks are utilized, one-directional information flow is assumed. The output signal of each neuron is transmitted to several neurons of the following layer by synaptical connections.

3.1. Training Procedure. The ANN with the presented structure is provided by a procedure for the identification of an optimal ANN architecture. The optimization algorithm analyses various configurations of hidden layers, especially their number as well as the number of neurons within each layer, and identifies an architecture providing best fitting to

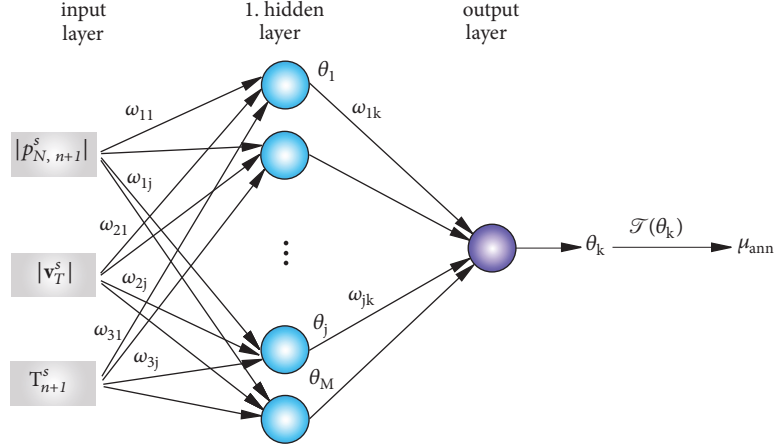


FIGURE 7: Architecture of an ANN applied for the computation of the friction coefficient μ_{ann} .

the experimental results. The ANN input signals x_i , $i = 1, 2, 3$, which enter the training procedure, are normalized

$$|x_i^s| = S(|x_i|) = \frac{|x_i| - \mu_{x_i}}{\sigma_{x_i}}. \quad (6)$$

Thereby, $x_i \in [p_N, v_T, T]$ are the contact pressures, sliding velocities and temperatures evaluated within the friction test. The function S accounts for the Gaussian distribution of experimental x_i magnitudes, where μ_{x_i} stands for the mean value and σ_{x_i} for the standard deviation.

The training of the feedforward ANN is executed by means of a backpropagation algorithm [11]. This method computes the gradient of an error function, which is defined as a difference between the experimental friction coefficient and the one provided by ANN, to identify the optimal configuration of synaptical weights ω_{ij} .

3.2. Computation of the Friction Coefficient by the ANN. A fitted ANN is applied within a friction law and enables the computation of the friction coefficient, which succeeds in several steps. First, the outcome of the hidden layer neuron θ_j is calculated for the normalized ANN input quantities $p_{N, n+1}^s$, v_T^s and T_{n+1}^s , which stem from the FE analysis

$$\begin{aligned} \theta_j &= \mathcal{F} \left(\sum_{i=1}^3 \omega_{ij} x_i + \nu_j \right) \\ &= \mathcal{F} \left(\omega_{1j} |p_{N, n+1}^s| + \omega_{2j} \|v_T^s\| + \omega_{3j} T_{n+1}^s + \nu_j \right). \end{aligned} \quad (7)$$

Thereby, ω_{ij} stands for the synaptical weight associated with a neural connection between the input layer neuron i and the hidden layer neuron j , whereas ν_j defines a bias term of the hidden layer neuron. Second, the outcome of the output layer neuron θ_k is calculated

$$\theta_k = \mathcal{F} \left(\sum_{j=1}^M \omega_{jk} \theta_j + \nu_k \right), \quad (8)$$

where ω_{jk} is the synaptical weight of the neural connection between the hidden layer neuron j and the output layer neuron k and ν_k is the bias term of the output layer neuron accordingly.

Finally, the output layer signal θ_k is evaluated by the transformation function \mathcal{F} to provide the friction coefficient μ_{ann}

$$\mu_{ann} = \mathcal{F}(\theta_k) = k_{sc} \operatorname{arctanh}(2\theta_k - 1). \quad (9)$$

Thereby, k_{sc} is the scale factor of the friction coefficient, which is calculated during the training procedure. The transformation function \mathcal{F} aims to scale the normalized signals processed within the ANN back to the real output parameter range.

The friction coefficient μ_{ann} computed by the ANN is evaluated within the friction law presented in (1), which is applied in the finite element simulation framework

$$\mathbf{t}_T = \mu_{ann}(|p_N|, \|v_T\|, T) |p_N| \frac{\mathbf{g}_T}{\|\mathbf{g}_T\|}. \quad (10)$$

4. Temperature-Dependent Friction Law Based on ANN

4.1. Friction in the Contact Formulation of the FEM. In the FEM, formulations for the tangential contact and friction are employed when two bodies slide on each other. Thereby, a stick and slip state as well as a transition zone between these states is considered in the tangential contact; see Figure 8. Especially, the slip state occurs if, due to the acting tangential force F_H , the critical tangential stress is achieved. In order to enable a smooth transition between the stick and slip state and to avoid differentiability problems in this zone, the hyperbolic tangent regularization is applied; see Figure 8. Thus, (10) results in

$$\mathbf{t}_T = \mu_{ann}(|p_N|, \|v_T\|, T) |p_N| \tanh \left(\frac{\|\Delta \mathbf{g}_T\|}{\beta} \right) \frac{\Delta \mathbf{g}_T}{\|\Delta \mathbf{g}_T\|}. \quad (11)$$

The hyperbolic tangent regularization, marked by a dashed line in Figure 8, is specified in dependency on the tangential

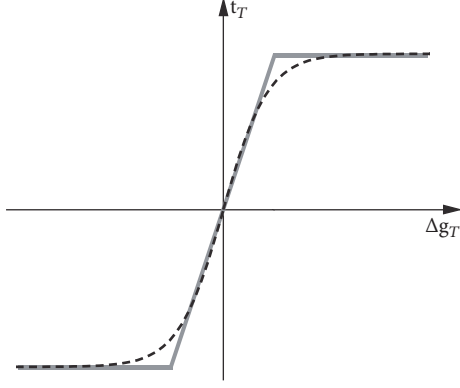


FIGURE 8: Regularization of the stick-slip relation.

slip rate $\Delta \mathbf{g}_T$ and the regularization coefficient β ; see [20]. Especially, while only minor relative slips are regarded during the sticking state, the magnitude of relative motion increases with the increase of the tangential stress \mathbf{t}_T until the critical \mathbf{t}_T and the slip state is reached. How close the regularized stick-slip transition approximates the classical double-branch transition depends on the magnitude of the coefficient β .

If in the FE analysis contact between two bodies is established at some position, the associated contact constraint is set active and the following penalty term is added to the expression for the total energy

$$\Pi_c = \frac{1}{2} \int_{\Gamma_c} (\epsilon_N (g_N)^2 + \mathbf{t}_T \cdot \mathbf{g}_T) dA, \quad \epsilon_N > 0. \quad (12)$$

Thereby, g_N denotes the penetration function within the normal contact formulation and Γ_c is the surface boundary in contact. In (12), $\epsilon_N = 10 \cdot k_{rep}$ stands for the penalty term and k_{rep} is the representative underlying element stiffness.

The minimization of the total energy, which leads to the weak form of the balance of momentum, requires the variation of (12)

$$\delta \Pi_c = \int_{\Gamma_c} (\epsilon_N g_N \delta g_N + \mathbf{t}_T \cdot \delta \mathbf{g}_T) dA, \quad \epsilon_N > 0. \quad (13)$$

Finally, the linearization of the weak form of balance of momentum needs to be provided for the Newton method to find the solution of the system of equations and, thus, a formulation for the linearized contact contribution is specified

$$d\delta \Pi_c = \int_{\Gamma_c} (dg_N \epsilon_N \delta g_N + d\delta g_N \epsilon_N g_N + d\mathbf{t}_T \cdot \delta \mathbf{g}_T + d\delta \mathbf{g}_T \cdot \mathbf{t}_T) dA. \quad (14)$$

Equation (14) contains a term defining the linearized tangential stress $d\mathbf{t}_T$, which is computed considering the developed constitutive equation for friction. Especially, since in (11) the tangential stresses are specified in dependency on the contact pressure, sliding velocity and temperature, the expression $d\mathbf{t}_T$

is a function of the derivatives of tangential stresses \mathbf{t}_T with respect to p_N , \mathbf{g}_T , and T

$$d\mathbf{t}_T = \frac{\partial \mathbf{t}_T}{\partial p_N} dp_N + \frac{\partial \mathbf{t}_T}{\partial \mathbf{g}_T} d\mathbf{g}_T + \frac{\partial \mathbf{t}_T}{\partial T} dT. \quad (15)$$

The formulation of these derivatives is given in the following section.

4.2. Linearization Terms for the Implementation of the Friction Law within the FEM. The implementation of the developed ANN based friction law within the framework of the FE analysis requires the specification of the algorithmic tangent including the derivatives of the tangential stresses $\partial \mathbf{t}_T / \partial p_N$, $\partial \mathbf{t}_T / \partial \mathbf{g}_T$, and $\partial \mathbf{t}_T / \partial T$.

The tangential stresses and their derivatives are computed in each time step t_{n+1} of the FE analysis based on the developed friction law. Thus, (11) results in

$$\mathbf{t}_{T,n+1} = \mu_{ann}(|p_{N,n+1}|, \|\mathbf{v}_T\|, T_{n+1}) \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right) \cdot |p_{N,n+1}| \frac{\Delta \mathbf{g}_T}{\|\Delta \mathbf{g}_T\|}. \quad (16)$$

For a simplified formulation of the derivatives of tangential stresses $\mathbf{t}_{T,n+1}$, the following notation of (16) is introduced:

$$\mathbf{t}_{T,n+1} = C_{fr,n+1} \mathbf{n}_{n+1}, \quad (17)$$

with $\mathbf{n}_{n+1} = \Delta \mathbf{g}_T / \|\Delta \mathbf{g}_T\|$ and C_{fr} defined as

$$C_{fr,n+1} = \mu_{ann}(|p_{N,n+1}|, \|\mathbf{v}_T\|, T_{n+1}) \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right) \cdot |p_{N,n+1}|. \quad (18)$$

First, the derivative of the tangential stress $\mathbf{t}_{T,n+1}$ with respect to the contact pressure $p_{N,n+1}$ is specified as

$$\frac{\partial \mathbf{t}_{T,n+1}}{\partial p_{N,n+1}} = \frac{\partial C_{fr,n+1}}{\partial |p_{N,n+1}|} \frac{\partial |p_{N,n+1}|}{\partial p_{N,n+1}} \mathbf{n}_{n+1}, \quad (19)$$

with

$$\frac{\partial C_{fr,n+1}}{\partial |p_{N,n+1}|} = \frac{\partial \mu_{ann}}{\partial |p_{N,n+1}|} \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right) |p_{N,n+1}| + \mu_{ann} \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right), \quad (20)$$

$$\frac{\partial |p_{N,n+1}|}{\partial p_{N,n+1}} = \text{sign}(p_{N,n+1}) = -1 \quad \text{for } p_{N,n+1} < 0. \quad (21)$$

Please note that, in (20), the derivative $\partial \mu_{ann} / \partial |p_{N,n+1}|$ is computed taking into account the architecture and the properties of the underlying ANN, which is presented later in this section.

Second, the derivative of tangential stress $\mathbf{t}_{T,n+1}$ with respect to the tangential slip $\mathbf{g}_{T,n+1}$ is formulated as

$$\frac{\partial \mathbf{t}_{T,n+1}}{\partial \mathbf{g}_{T,n+1}} = \mathbf{n}_{n+1} \otimes \frac{\partial C_{fr,n+1}}{\partial \mathbf{g}_{T,n+1}} + C_{fr,n+1} \frac{\partial \mathbf{n}_{T,n+1}}{\partial \mathbf{g}_{T,n+1}}, \quad (22)$$

where

$$\begin{aligned} \frac{\partial C_{fr,n+1}}{\partial \mathbf{g}_{T,n+1}} &= \frac{\partial \mu_{ann}}{\partial \mathbf{g}_{T,n+1}} \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right) |p_{N,n+1}| \\ &+ \frac{\partial (\tanh(\|\Delta \mathbf{g}_T\|/\beta))}{\partial \mathbf{g}_{T,n+1}} \mu_{ann} |p_{N,n+1}|. \end{aligned} \quad (23)$$

The formulation of the derivative of the friction coefficient μ_{ann} with respect to the tangential slip $\mathbf{g}_{T,n+1}$ in (23) is based on the formulation of the sliding velocity \mathbf{v}_T

$$\frac{\partial \mu_{ann}}{\partial \mathbf{g}_{T,n+1}} = \frac{\partial \mu_{ann}}{\partial \|\mathbf{v}_T\|} \frac{\partial \|\mathbf{v}_T\|}{\partial \mathbf{g}_{T,n+1}}. \quad (24)$$

Thereby, the derivative $\partial \mu_{ann} / \partial \|\mathbf{v}_T\|$ is computed considering the differentiability and the architecture of the ANN.

Taking into account the dependency between the sliding velocity $\|\mathbf{v}_T\|$ and the tangential slip $\mathbf{g}_{T,n+1}$, $\|\mathbf{v}_T\| = \|\Delta \mathbf{g}_T\| / \Delta t$, where $\Delta \mathbf{g}_T = \mathbf{g}_{T,n+1} - \mathbf{g}_{T,n}$, yields

$$\frac{\partial \|\mathbf{v}_T\|}{\partial \mathbf{g}_{T,n+1}} = \frac{\partial \|\mathbf{v}_T\|}{\partial \Delta \mathbf{g}_T} \cdot \frac{\partial \Delta \mathbf{g}_T}{\partial \mathbf{g}_{T,n+1}} = \frac{1}{\Delta t} \frac{\Delta \mathbf{g}_T}{\|\Delta \mathbf{g}_T\|} \cdot \mathbf{I}, \quad (25)$$

where \mathbf{I} is the second order identity tensor.

To complete the formulation in (23), the derivative of the hyperbolic tangent regularization term is specified

$$\begin{aligned} &\frac{\partial (\tanh(\|\Delta \mathbf{g}_T\|/\beta))}{\partial \mathbf{g}_{T,n+1}} \\ &= \left(1 - \tanh^2\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right)\right) \frac{1}{\beta} \frac{\Delta \mathbf{g}_T}{\|\Delta \mathbf{g}_T\|} \cdot \mathbf{I}. \end{aligned} \quad (26)$$

The last term required for the specification of the derivative $\partial \mathbf{t}_{T,n+1} / \partial \mathbf{g}_{T,n+1}$ in (22) is given by

$$\frac{\partial \mathbf{n}_{n+1}}{\partial \mathbf{g}_{T,n+1}} = \frac{\partial \mathbf{n}_{n+1}}{\partial \Delta \mathbf{g}_T} \cdot \frac{\partial \Delta \mathbf{g}_T}{\partial \mathbf{g}_{T,n+1}} = \frac{\partial \mathbf{n}_{n+1}}{\partial \Delta \mathbf{g}_T} \cdot \mathbf{I}, \quad (27)$$

where

$$\frac{\partial \mathbf{n}_{n+1}}{\partial \Delta \mathbf{g}_T} = \frac{1}{\|\Delta \mathbf{g}_T\|} (\mathbf{I} - \mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1}). \quad (28)$$

Finally, the expression for the derivative of the tangential stress $\mathbf{t}_{T,n+1}$ with respect to the temperature T_{n+1} is given, as a last part of the formulation of the algorithmic tangent

$$\frac{\partial \mathbf{t}_{T,n+1}}{\partial T_{n+1}} = \frac{\partial C_{fr,n+1}}{\partial T_{n+1}} \mathbf{n}_{n+1}, \quad (29)$$

where

$$\frac{\partial C_{fr,n+1}}{\partial T_{n+1}} = \frac{\partial \mu_{ann}}{\partial T_{n+1}} \tanh\left(\frac{\|\Delta \mathbf{g}_T\|}{\beta}\right) |p_{N,n+1}|. \quad (30)$$

The introduced equations specifying the algorithmic tangent require the computation of the derivatives $\partial \mu_{ann} / \partial |p_{N,n+1}|$, $\partial \mu_{ann} / \partial \|\mathbf{v}_T\|$, and $\partial \mu_{ann} / \partial T_{n+1}$, which refer to the formulation of the friction coefficient computed by the ANN. Especially, the calculation of these derivatives is conditioned by the differentiability of the ANN formulation as well as of the applied activation function. In this work, sigmoid logistic activation functions are utilized, for which the derivative equals to

$$\mathcal{F}' = \mathcal{F} (1 - \mathcal{F}). \quad (31)$$

From the formulation of the friction coefficient given in (9) and from the definition of the neuronal output shown in (7) and (8), the derivative $\partial \mu_{ann} / \partial |p_{N,n+1}|$ is specified as

$$\frac{\partial \mu_{ann}}{\partial |p_{N,n+1}|} = \frac{\partial \mu_{ann}}{\partial \theta_k} \frac{\partial \theta_k}{\partial |p_{N,n+1}^s|} \frac{\partial |p_{N,n+1}^s|}{\partial |p_{N,n+1}|}, \quad (32)$$

with

$$\begin{aligned} \frac{\partial \theta_k}{\partial |p_{N,n+1}^s|} &= \frac{\partial \theta_k}{\partial \theta_j} \frac{\partial \theta_j}{\partial |p_{N,n+1}^s|} = \mathcal{F}' \left(\sum_{j=1}^M w_{jk} \theta_j + \nu_k \right) \\ &\cdot \mathcal{F}' (w_{1j} |p_{N,n+1}^s| + w_{2j} \|\mathbf{v}_T\| + w_{3j} T_{n+1} + \nu_j). \end{aligned} \quad (33)$$

Considering (31), the expression in (33) can be rewritten as

$$\frac{\partial \theta_k}{\partial |p_{N,n+1}^s|} = \theta_k (1 - \theta_k) \sum_{j=1}^M w_{jk} \theta_j (1 - \theta_j) w_{1j}. \quad (34)$$

In a similar way, the derivative $\partial \mu_{ann} / \partial \|\mathbf{v}_T\|$ is formulated

$$\frac{\partial \mu_{ann}}{\partial \|\mathbf{v}_T\|} = \frac{\partial \mu_{ann}}{\partial \theta_k} \frac{\partial \theta_k}{\partial \|\mathbf{v}_T^s\|} \frac{\partial \|\mathbf{v}_T^s\|}{\partial \|\mathbf{v}_T\|}, \quad (35)$$

where

$$\frac{\partial \theta_k}{\partial \|\mathbf{v}_T^s\|} = \theta_k (1 - \theta_k) \sum_{j=1}^M w_{jk} \theta_j (1 - \theta_j) w_{2j}. \quad (36)$$

The derivative $\partial \mu_{ann} / \partial T_{n+1}$ is specified analogously to the derivatives $\partial \mu_{ann} / \partial |p_{N,n+1}|$ and $\partial \mu_{ann} / \partial \|\mathbf{v}_T\|$

$$\frac{\partial \mu_{ann}}{\partial T_{n+1}} = \frac{\partial \mu_{ann}}{\partial \theta_k} \frac{\partial \theta_k}{\partial T_{n+1}^s} \frac{\partial T_{n+1}^s}{\partial T_{n+1}}, \quad (37)$$

where

$$\frac{\partial \theta_k}{\partial T_{n+1}^s} = \theta_k (1 - \theta_k) \sum_{j=1}^M w_{jk} \theta_j (1 - \theta_j) w_{3j}. \quad (38)$$

For the formulation of the derivatives in (32), (35), and (37), terms accounting for the transformation function \mathcal{T} and the normalization function S are defined as

$$\frac{\partial \mu_{am}}{\partial \theta_k} = \frac{\partial \mathcal{T}(\theta_k)}{\partial \theta_k} = \frac{-1}{2(\theta_k^2 - \theta_k)}, \quad (39)$$

$$\frac{\partial |p_{N,n+1}^s|}{\partial |p_{N,n+1}|} = \frac{\partial S(|p_{N,n+1}|)}{\partial |p_{N,n+1}|} = \frac{1}{\sigma_{p_N}}, \quad (40)$$

$$\frac{\partial \|\mathbf{v}_T^s\|}{\partial \|\mathbf{v}_T\|} = \frac{\partial S(\|\mathbf{v}_T\|)}{\partial \|\mathbf{v}_T\|} = \frac{1}{\sigma_{v_T}},$$

$$\frac{\partial T_{n+1}^s}{\partial T_{n+1}} = \frac{\partial S(T_{n+1})}{\partial T_{n+1}} = \frac{1}{\sigma_T}. \quad (41)$$

5. FE Frictional Contact Simulation of Elastomeric Structures

The developed metamodel based friction formulation is applied for the investigation of contact properties of elastomeric structures and elements. Three examples are prepared. First, an isothermal FE simulation of a rubber block sliding on a rigid surface is carried out, whereas various rubber block temperatures are applied to assess their influence on the tangential stresses in the contact interface. Second, a fully thermo-mechanically coupled FE analysis of a rubber block with tread pattern moving on a tire-rotation-like displacement path is accomplished to quantify the performance of the friction law in a thermo-mechanically coupled simulation. Finally, isothermal FE analyses of a tire in a steady state rolling situation are carried out to visualize the applicability of the friction law within the steady state transport analysis, which is based on the arbitrary Lagrangian-Eulerian approach.

In all three examples, the constitutive behaviour of rubber is described by a hyperelastic material formulation utilizing the Marlow model; see [21]. The strain energy density is determined in the Marlow model by

$$W(I_1) = \int_0^{\lambda_T(I_1)-1} T(\epsilon) d\epsilon, \quad (42)$$

where I_1 is the first invariant of the left Cauchy-Green strain tensor $\mathbf{B} = \mathbf{F}\mathbf{F}^T$ and \mathbf{F} is the deformation gradient. In (42), ϵ denotes the uniaxial strain, $T(\epsilon)$ the nominal uniaxial traction, and $\lambda_T(I_1)$ stands for the uniaxial stretch obtained as a solution of the following equation

$$\lambda_T(I_1)^3 - I_1 \lambda_T(I_1) + 2 = 0. \quad (43)$$

In order to define $T(\epsilon)$, a uniaxial test data set is applied. The Marlow model is selected for the implementation within the following examples due to its straightforward formulation by means of only the first invariant I_1 . Several hyperelastic material models, as the Yeoh model [22] or the Arruda & Boyce model [23], are specified using only the first invariant I_1 as well, though they are characterized by a certain mathematical formulation with various parameters, which need to be fitted. Contrarily, in the Marlow model the material

TABLE 1: Loading and boundary conditions in the simulation of a sliding rubber block.

$p_N = 0.248 \text{ MPa}$		$p_N = 0.548 \text{ MPa}$	
$\mathbf{v}_T = 70 \text{ mm/s}$		$\mathbf{v}_T = 70 \text{ mm/s}$	
Sim	$T[^\circ\text{C}]$	Sim	$T[^\circ\text{C}]$
1	20	4	20
2	70	5	70
3	100	6	100

response is determined solely by the test data in form of uniaxial stress-strain curve, which integration yields directly the strain energy density, as shown in equation (42). The idea of modelling rubber as a viscoelastic material, which is presented, e.g., in the model by Simo [24], Reese & Govindjee [25], or Holzapfel [26], is not followed in this contribution but could be introduced within the following examples without any restrictions.

In all examples, the artificial neural network presented in Sections 2 and 3 with an architecture comprised of 3 input layer neurons, 14 hidden layer neurons, and 1 output layer neuron is applied.

5.1. Rubber Block Sliding on a Rigid Surface. Within the first example, a rubber block sliding on a fixed rigid surface is investigated. The rubber block is subjected to a normal pressure p_N applied uniformly at the top of the block and moves with a constant velocity \mathbf{v}_T ; see Figure 9. The rubber block has a prescribed initial temperature T , which remains constant during the simulation due to the isothermal conditions within the simulation.

In this example, two normal pressure magnitudes $p_N = 0.248 \text{ MPa}$ and $p_N = 0.548 \text{ MPa}$ are investigated; see Table 1, whereas for each p_N three rubber block temperatures $T = [20, 70, 100]^\circ\text{C}$ are prescribed. Thereby, in all six simulations, a sliding velocity $\mathbf{v}_T = 70 \text{ mm/s}$ is applied.

The rubber block is discretized by 8-node solid elements. The application of the newly developed friction law accounting for the dependency of μ on the nodal p_N , \mathbf{v}_T , and T magnitudes enables an adequate computation of the tangential stresses.

In Figure 9, the tangential stresses \mathbf{t}_T obtained in all six simulations are plotted against the tangential slip \mathbf{g}_T . Thereby, in each graph, the sticking regime with minor elastic slip magnitudes, the transition zone between sticking and sliding and the sliding regime with a constant \mathbf{t}_T magnitude are clearly visible. Furthermore, a significant influence of the temperature-dependent formulation of the friction law on the course of tangential stresses is apparent from Figure 9. Especially, at $p_N = 0.248 \text{ MPa}$ as well as at $p_N = 0.548 \text{ MPa}$, \mathbf{t}_T decreases with increasing temperature T , which is conform with experimental results presented in Figure 1. This simple example shows that sticking, sliding, and the transition between these states can be simulated properly with the developed formulation of the friction law.

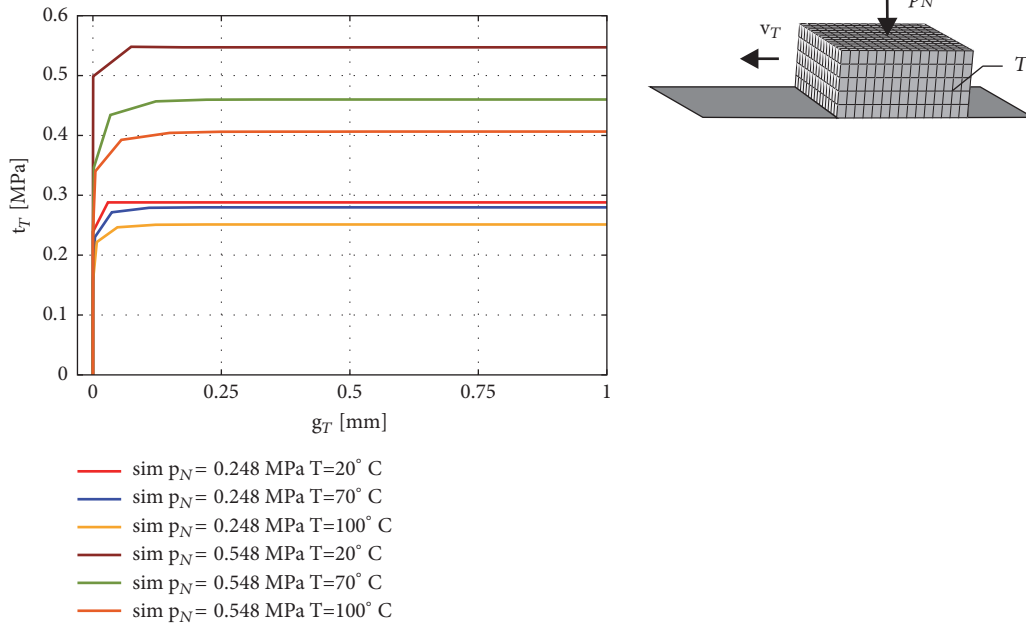


FIGURE 9: Tangential stresses in the rubber block simulation in dependency on temperature and normal pressure.

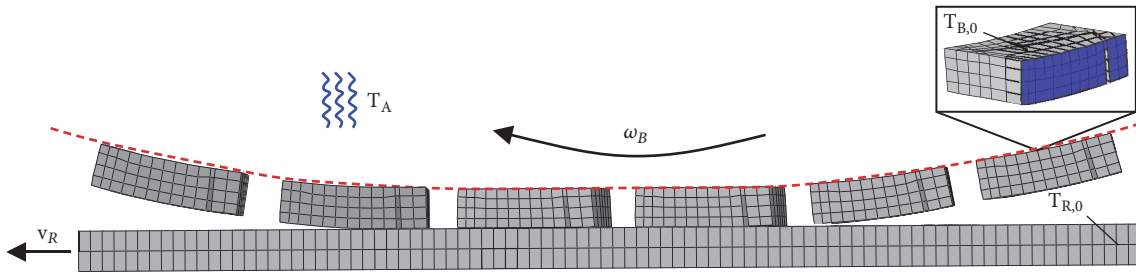


FIGURE 10: Rubber block moving on a tire-rotation-like displacement path.

5.2. Rubber Block Moving on a Tire-Rotation-Like Displacement Path. In the second example, the proposed friction formulation is applied to a thermo-mechanically coupled FE analysis of a rubber block moving on a tire-rotation-like displacement path. In Figure 10, the considered rubber block with tread pattern is visualized. Furthermore, the movement of the block along a displacement path is presented, which consists of three steps including the establishment of contact between the rubber block and the road, the sliding of the block at the road surface, and the final step, where the block detaches from the road surface.

A tire in a standard operating condition is subjected to angular and translational velocity, whereas the magnitudes of these velocities determine whether braking, acceleration or a free rolling state is obtained. Analogously, these states can be achieved in the presented simulation of the rubber block by applying a translation velocity v_R to the road and letting the block move on a displacement path according to a prescribed angular velocity ω_B ; see Figure 10.

In the applied thermo-mechanically coupled simulation, two heat sources can be identified. First, the temperature

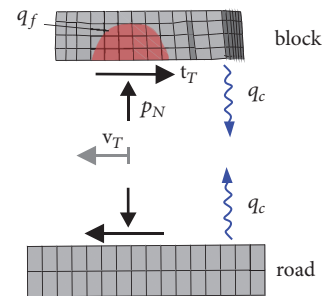


FIGURE 11: Rubber block-road contact interface.

increase in the contact interface due to friction energy dissipation is considered. Second, the heat exchange between the rubber block and surrounding air influences the rubber block temperature; see Figure 11. The prescribed initial temperatures of the rubber block and the road are denoted by $T_{B,0}$ and $T_{R,0}$, respectively, whereas the air temperature is denoted by T_A ; compare Figure 10.

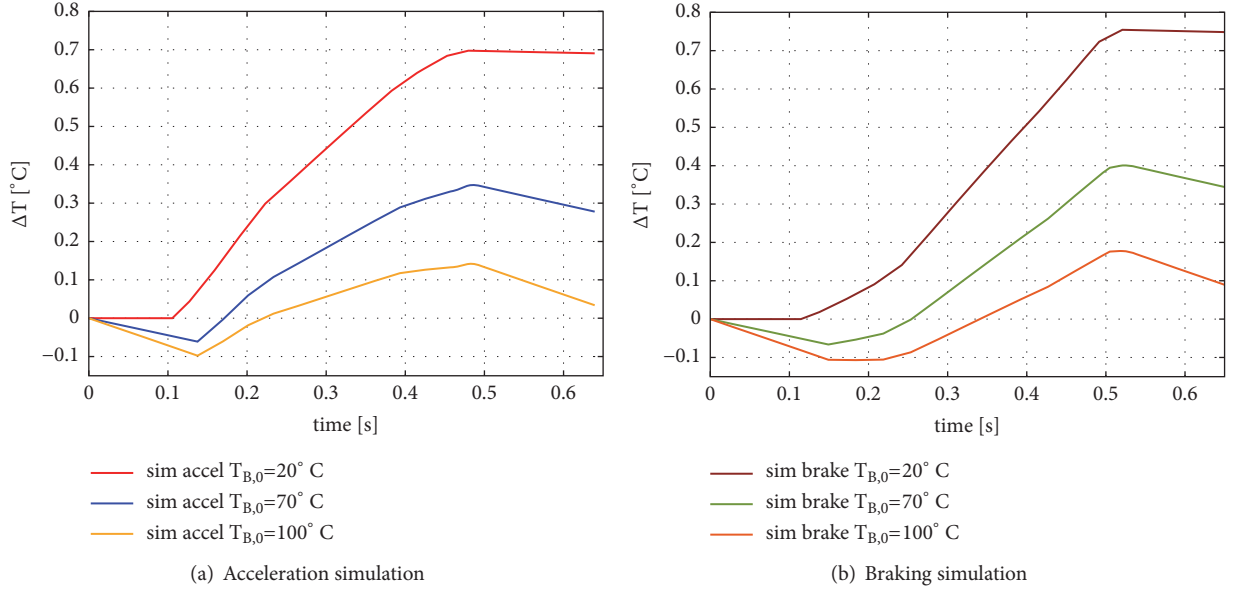


FIGURE 12: Temperature amplitude in the acceleration and braking simulation.

The heat flux q_f arising due to the friction energy dissipation in the contact interface is defined as

$$q_f = \eta f \mathcal{D}_s, \quad (44)$$

where \mathcal{D}_s is the friction energy, η is the fraction of \mathcal{D}_s converted into heat, and f is the heat distribution factor between the two surfaces in contact. While η and f take values from the interval $[0, 1]$, the friction energy \mathcal{D}_s is given as

$$\mathcal{D}_s = \mathbf{t}_T \dot{\mathbf{g}}_T. \quad (45)$$

The heat flux q_c arising due to the heat exchange between the rubber block and the surrounding air is defined as

$$q_c = h(T_B - T_A), \quad (46)$$

where h stands for the film coefficient. The left and right side as well as the bottom of the block, marked by blue shaded surface in Figure 10, are subjected to the heat exchange with the surrounding air. The remaining surfaces of the block are assumed to be in contact with other tire parts.

Both, the rubber block and the road, are modelled as flexible bodies, which are discretized by 8-node solid elements with displacements and temperature degrees of freedom. The road is modelled as a linear elastic material with temperature-dependent Young's modulus. The applied magnitudes of the thermal conductivity k , the specific heat c_p and the Young's modulus E are given in Table 2.

In this example, two types of simulation are performed, the first one capturing the accelerating and the second one the braking condition. These conditions are defined by the relation of the angular and the translational velocity, compare Table 3. Furthermore, for each simulation type, three initial rubber block temperatures $T_{B,0} = [20, 70, 100]^\circ\text{C}$ are considered. The initial road temperature is set to $T_{R,0} =$

TABLE 2: Material properties in the simulation of a rubber block moving on a tire-rotation-like displacement path.

	rubber material	road material
k [W/mK]	0.279	0.9
c_p [J/kgK]	$2010 \cdot 10^6$	$1200 \cdot 10^6$
E [MPa]	-	2200 at $T_R = 20^\circ\text{C}$
	-	1300 at $T_R = 100^\circ\text{C}$
	-	530 at $T_R = 200^\circ\text{C}$

20°C and the air temperature to $T_A = 20^\circ\text{C}$ in all simulations. Further quantities required for the computation of heat fluxes are defined as $\eta = 1$, $h = 22 \cdot 10^{-1} \text{W/mm}^2\text{K}$, $f = 0.8$, whereas f denotes the heat distribution factor determining the heat flux into the block surface.

In Figures 12 and 13, the results of the accelerating and braking simulation are plotted for a chosen node in the bottom surface of the rubber block, which is contacting the road. In Figure 12, the temperature amplitude $\Delta T = T_{B,i} - T_{B,0}$, defined as a difference between the nodal temperature $T_{B,i}$ in the i th time increment and the initial temperature $T_{B,0}$ is plotted over the simulation time. Thereby, in Figure 12(a) the results of the accelerating simulation and in Figure 12(b) the results of the braking simulation performed at various initial rubber block temperatures $T_{B,0}$ are visualized.

Within Figure 12(a), the three stages of the simulation can be clearly identified, first stage—before the contact establishment—when the nodal temperature decreases due to the heat exchange with air, the full contact stage when the temperature increases as a result of the friction energy dissipation, and the third stage when the block leaves the contact interface and the temperature decreases again due to heat exchange with air. It should be noticed that, for the simulation with initial block temperature $T_{B,0} = 20^\circ\text{C}$, the

TABLE 3: Loading and boundary conditions in the simulation of a rubber block moving on a tire-rotation-like displacement path.

accelerating				braking			
$\mathbf{v}_R = 300 \text{ mm/s}$				$\mathbf{v}_R = 300 \text{ mm/s}$			
$\omega_B = 1.064 \text{ rad/s}$				$\omega_B = 0.98 \text{ rad/s}$			
Sim	$T_{B,0} [^\circ\text{C}]$	$p_N [N/mm^2]$	$\mathbf{v}_T [mm/s]$	Sim	$T_{B,0} [^\circ\text{C}]$	$p_N [N/mm^2]$	$\mathbf{v}_T [mm/s]$
1	20	0.506	87.438	4	20	0.487	78.197
2	70	0.502	81.038	5	70	0.492	78.748
3	100	0.500	81.042	6	100	0.491	78.750

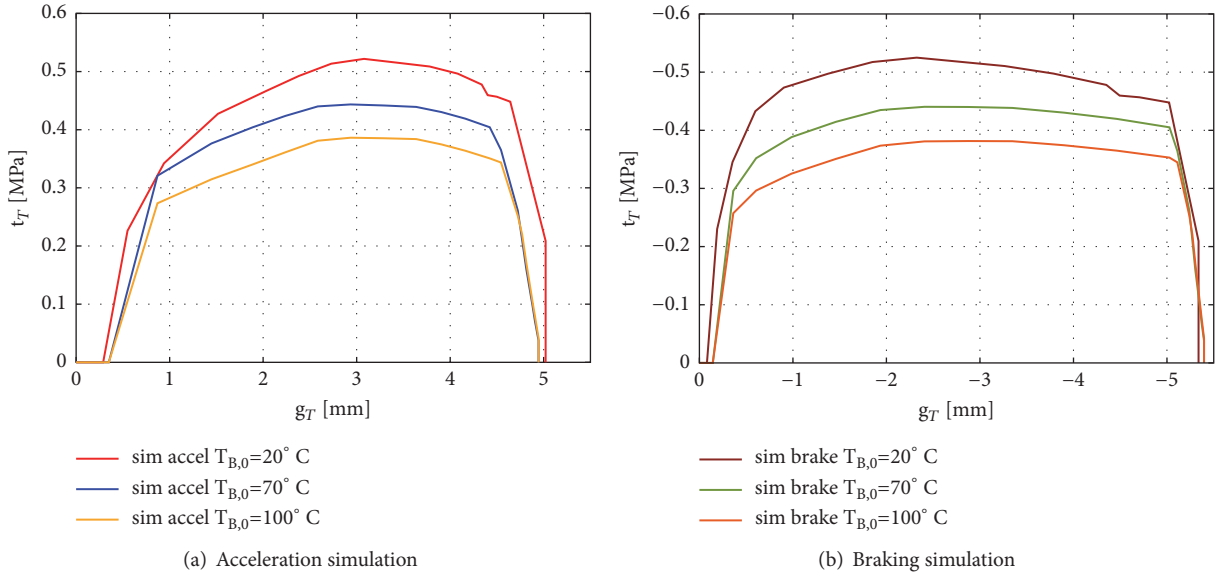


FIGURE 13: Tangential stresses versus slip in the acceleration and braking simulation.

temperature is not decreasing in stage one but stays constant due to the same magnitude of the air temperature $T_{A,0} = 20^\circ\text{C}$. The temperature increase within the simulation with $T_{B,0} = 20^\circ\text{C}$ is more pronounced than for simulations with $T_{B,0} = 70^\circ\text{C}$ and $T_{B,0} = 100^\circ\text{C}$ due to higher values of the friction coefficient μ and, therefore, higher magnitude of the dissipated friction energy \mathcal{D}_s . The same observations are valid for the results of the braking simulation presented in Figure 12(b). In this example, larger temperature amplitudes are obtained within the braking simulation than within the accelerating simulation i.a. due to a slightly larger magnitude of relative braking in comparison to relative acceleration.

In Figure 13, the tangential stresses t_T are plotted versus the tangential slip g_T . Thereby, a clear tendency characterized by achieving higher t_T at lower initial rubber block temperatures $T_{B,0}$ is visible, which is conform with the magnitudes of the friction coefficient obtained in the experiments for the contact pressure and sliding velocity ranges under consideration. Representative values for p_N and \mathbf{v}_T , which are obtained in the simulation during sliding at $g_T = \pm 3 \text{ mm}$, are shown in Table 3. Furthermore, within a single simulation at certain initial rubber block temperature $T_{B,0}$, a variation of tangential stress magnitudes in the full contact stage can be observed. This effect is achieved due to the

TABLE 4: Residual force in subsequent equilibrium iterations at two selected time increments t_1 and t_2 .

iteration	t_1	t_2
	$F_{res} [N]$	$F_{res} [N]$
1	$1.406 \cdot 10^{-3}$	$-1.545 \cdot 10^{-3}$
2	$9.356 \cdot 10^{-6}$	$4.492 \cdot 10^{-6}$

dependency of the tangential stress formulation on the local nodal pressure and temperature, which magnitudes change during the simulation.

Within this example, the rate of convergence of the formulation is examined. In almost all time increments the Newton method is able to find the solution within 1-3 iterations. Thereby, a quadratic rate of convergence is obtained, which is confirmed by the magnitudes of the residual force F_{res} in subsequent equilibrium iterations at two selected time increments t_1 and t_2 ; see Table 4. Thereby, the considered time increments refer to the time, when the rubber block is sliding on the road surface.

The presented results confirm that the application of the newly developed friction law within a thermo-mechanically

TABLE 5: Representative values of tangential stresses obtained in the FE tire simulation with corresponding p_N , v_T , and T values.

node	T [°C]	p_N [N/mm ²]	v_T [mm/s]	t_T [N/mm ²]
1	20	0.371	147.887	-0.355
1	70	0.384	181.090	-0.362
2	20	0.455	130.700	-0.396
2	70	0.467	179.130	-0.402
3	20	0.448	161.790	-0.426
3	70	0.448	514.825	-0.423

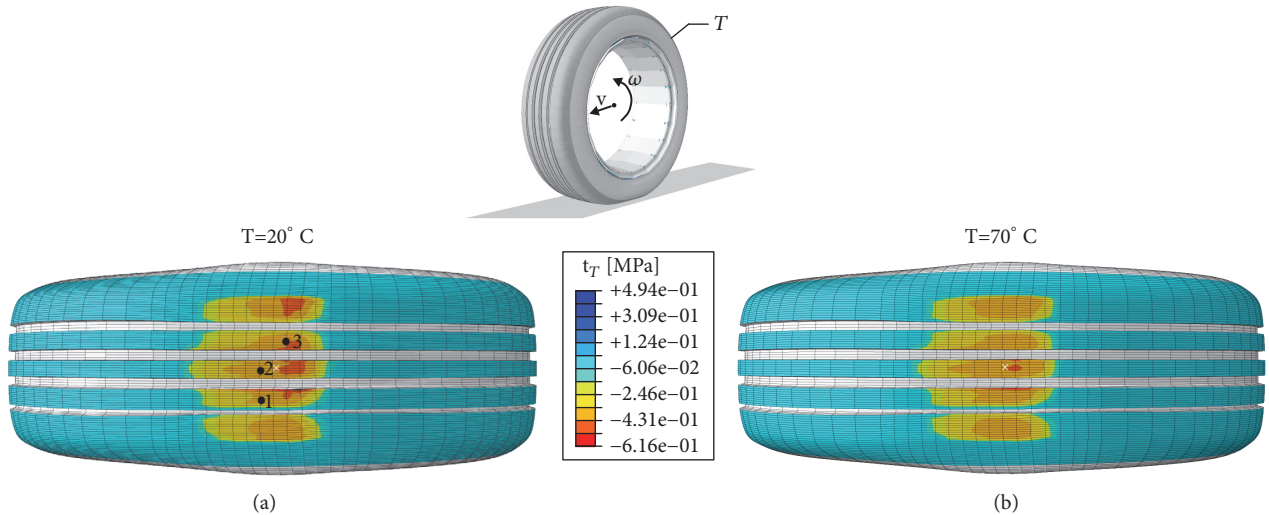


FIGURE 14: Tangential stress distribution in the contact interface of a rolling tire.

coupled FE simulation enables to properly capture the contact phenomena related to temperature and friction.

5.3. Tire Simulation. In this example, two isothermal FE simulations of tires in the steady state rolling condition are performed. In the first simulation, the temperature of all tire components is set to $T = 20^\circ\text{C}$, whereas in the second simulation to $T = 70^\circ\text{C}$. The tire is subjected to inner pressure $p = 0.26 \text{ MPa}$ and a vertical load $F_V = 5250 \text{ N}$. In both simulations, the translational velocity is equal to $v = 108 \text{ km/h}$ and the angular velocity to $\omega = 100 \text{ rad/s}$, which corresponds to a braking state; see Figure 14.

In the applied tire model, the rubber parts are discretized by 8-node hybrid solid elements, in which pressure stress is an additional independently interpolated solution variable, that is coupled to the displacement solution. Such element formulation is suited for nearly incompressible materials. The tire reinforcement is modelled by means of rebar layers embedded in the solid elements and the road is defined as a rigid body.

The FE steady state rolling analysis applied in the framework of this contribution is accomplished by means of the arbitrary Lagrangian-Eulerian approach; see [27–29]. Thus, tire rotation is regarded as a rigid body rotation and described in the Eulerian frame whereas the material deformation is described in the Lagrangian frame. The rotation is considered as a material flow through a standing FE mesh, where the

flow occurs at surface streamlines. The application of the arbitrary Lagrangian-Eulerian approach permits an adaptive discretization of the tire model, where a fine mesh is required solely in the tire-road contact zone. Thus, efficient FE simulations characterized by a low computational cost but a high accuracy can be performed.

The application of the developed friction law within a steady state transport analysis, which is stationary and thus time independent, requires the consideration of a different dependency between the sliding velocity $\|\mathbf{v}_T\|$ and the tangential slip $\mathbf{g}_{T,n+1}$ than in a transient analysis. Since this dependency is given as $\|\mathbf{v}_T\| = \|\Delta\mathbf{g}_T\|$, in equation (25), which is utilized for the specification of the algorithmic tangent, the term $1/\Delta t$ vanishes.

The results of the FE tire simulations accomplished in this study are visualized in Figure 14. Especially, the distributions of tangential stresses obtained in the FE analyses with prescribed tire temperatures $T = 20^\circ\text{C}$ and $T = 70^\circ\text{C}$ are shown. Thereby, the stress component in the driving direction of the tire is considered. In Figure 14, the tangential stresses t_T vary with respect to the local temperature, contact pressure and sliding velocity. In Table 5, representative values of t_T and the corresponding influencing factors are given for selected nodes in the tire contact area, see Figure 14. The tangential stresses on the right-hand side of the contact patch are larger in the tire simulation with tire temperature $T = 20^\circ\text{C}$ than with tire temperature $T = 70^\circ\text{C}$.

This example confirms, that the developed metamodel based constitutive equation for friction is applicable within the arbitrary Lagrangian-Eulerian approach and can be utilized in the simulation of steady state rolling tires.

6. Conclusions

In this contribution, a new friction model enabling to capture the thermo-mechanical tribological properties of elastomeric structures is proposed. The developed macroscopic equation for friction accounts for the dependency of the friction coefficient on the local contact pressure, sliding velocity and temperature in the contact interface. Since a metamodel is integrated into the friction formulation, an extraordinary high fitting quality to the experimental results is achieved. Furthermore, a consistent formulation of the law enabling its implementation within the framework of the finite element method is provided. Various applications of the proposed constitutive equation for friction to FE simulations indicate its good performance in transient isotherm and thermo-mechanically coupled analyses as well as in isotherm steady state transport analyses.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) under grant no. KA 1163/30 within the DFG research unit FOR 2089.

References

- [1] A. Serafinska, N. Hassoun, and M. Kaliske, "Numerical optimization of wear performance - Utilizing a metamodel based friction law," *Computers & Structures*, vol. 165, pp. 10–23, 2016.
- [2] G. Heinrich and M. Klüppel, "Rubber friction, tread deformation and tire traction," *Wear*, vol. 265, no. 7-8, pp. 1052–1060, 2008.
- [3] A. Le Gal and M. Klüppel, "Investigation and modelling of rubber stationary friction on rough surfaces," *Journal of Physics: Condensed Matter*, vol. 20, no. 1, 2008.
- [4] B. N. J. Persson, "Contact mechanics for randomly rough surfaces," *Surface Science Reports*, vol. 61, no. 4, pp. 201–227, 2006.
- [5] P. Wriggers and J. Reinelt, "Multi-scale approach for frictional contact of elastomers on rough rigid surfaces," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 21-26, pp. 1996–2008, 2009.
- [6] K. Falk, R. Lang, and M. Kaliske, "Multiscale simulation to determine rubber friction on asphalt surfaces," *Tire Science and Technology*, vol. 44, no. 4, pp. 226–247, 2016.
- [7] F. Hartung, R. Kienle, T. Götz et al., "Numerical determination of hysteresis friction on different length scales and comparison to experiments," *Tribology International*, vol. 127, pp. 165–176, 2018.
- [8] T. Huemer, W. N. Liu, J. Eberhardsteiner, and H. A. Mang, "A 3D finite element formulation describing the frictional behavior of rubber on ice and concrete surfaces," *Engineering Computations*, vol. 18, no. 3-4, pp. 417–436, 2001.
- [9] A. R. Savkoor, "Some aspects of friction and wear of tyres arising from deformations, slip and stresses at the ground contact," *Wear*, vol. 9, no. 1, pp. 66–78, 1966.
- [10] R. van der Steen, *Enhanced friction modeling for steady-state rolling tires [Ph.D. thesis]*, Technische Universiteit Eindhoven, 2010.
- [11] S. Haykin, *Neural Networks: a Comprehensive Foundation*, Prentice-Hall, 1999.
- [12] H. Adeli, "Neural networks in civil engineering: 1989–2000," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126–142, 2001.
- [13] C. Zopf and M. Kaliske, "Numerical characterisation of uncured elastomers by a neural network based approach," *Computers & Structures*, vol. 182, pp. 504–525, 2017.
- [14] S. Freitag, *Modellfreie numerische Prognosemethoden zur Tragwerksanalyse [Dissertation, thesis]*, Technische Universität Dresden, 2010.
- [15] A. Chamekh, H. Bel Hadj Salah, and R. Hambli, "Inverse technique identification of material parameters using finite element and neural network computation," *The International Journal of Advanced Manufacturing Technology*, vol. 44, no. 1-2, pp. 173–179, 2009.
- [16] J. L. Rogers, "Simulating structural analysis with neural network," *Journal of Computing in Civil Engineering*, vol. 8, no. 2, pp. 252–265, 1994.
- [17] W. Graf, J.-U. Sickert, S. Pannier, and M. Kaliske, "Robust design with uncertain data and response surface approximation," in *Proceedings of the 4th International Workshop on Reliable Engineering Computing (REC 2010)*, M. Beer, R. L. Muhanna, and R. L. Mullen, Eds., pp. 554–573, Singapore, 2010.
- [18] W. Graf, J. Sickert, S. Freitag, S. Pannier, and M. Kaliske, "Neural network approaches in structural analysis considering imprecision and variability," in *Soft Computing Methods for Civil and Structural Engineering*, pp. 59–85, Saxe-Coburg Publications, Stirlingshire, UK, 2011.
- [19] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, USA, 1995.
- [20] P. Wriggers, *Computational Contact Mechanics*, Wiley, Chichester, 2002.
- [21] R. S. Marlow, "A general first-invariant hyperelastic constitutive model," in *Constitutive Models for Rubber III*, J. Busfield and A. Muhr, Eds., pp. 157–160, Balkema Publications, Lisse, 2003.
- [22] O. H. Yeoh, "Some forms of the strain energy function for rubber," *Rubber Chemistry and Technology*, vol. 66, no. 5, pp. 754–771, 1993.
- [23] E. M. Arruda and M. C. Boyce, "A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials," *Journal of the Mechanics and Physics of Solids*, vol. 41, no. 2, pp. 389–412, 1993.

- [24] J. C. Simo, "On a fully three-dimensional finite-strain viscoelastic damage model: Formulation and computational aspects," *Computer Methods in Applied Mechanics and Engineering*, vol. 60, no. 2, pp. 153–173, 1987.
- [25] S. Reese and S. Govindjee, "A theory of finite viscoelasticity and numerical aspects," *International Journal of Solids and Structures*, vol. 35, no. 26-27, pp. 3455–3482, 1998.
- [26] G. A. Holzapfel, "On large strain viscoelasticity: Continuum formulation and finite element applications to elastomeric structures," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 22, pp. 3903–3926, 1996.
- [27] T. J. Hughes, W. K. Liu, and T. Zimmermann, "Lagrangian-Eulerian finite element formulation for incompressible viscous flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 29, no. 3, pp. 329–349, 1981.
- [28] M. Kaliske, D. Zheng, M. Andre, and C. Bertram, "Efficient steady-state simulations up to high speed with dissipative tire characteristics," *Vehicle Systems Dynamics Journal Supplement*, vol. 40, pp. 175–194, 2003.
- [29] U. Nackenhorst, "The ALE-formulation of bodies in rolling contact: Theoretical foundations and finite element approach," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 39-41, pp. 4299–4322, 2004.

Research Article

A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition

Rubén Ibáñez,¹ Emmanuelle Abisset-Chavanne,¹ Amine Ammar ,² David González ,³ Elías Cueto ,³ Antonio Huerta,⁴ Jean Louis Duval,⁵ and Francisco Chinesta ¹

¹ESI Chair, ENSAM ParisTech. 151, bvd. de l'Hôpital, F-75013 Paris, France

²LAMPA, ENSAM ParisTech. 2, bvd. de Ronceray. F-49035 Angers, France

³Aragon Institute of Engineering Research, Universidad de Zaragoza. Maria de Luna, s.n. E-50018 Zaragoza, Spain

⁴Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya. Jordi Girona 1-3, E-08034 Barcelona, Spain

⁵ESI Group. Parc Icade, Immeuble le Seville, 3 bis, Saarinen, CP 50229, 94528, Rungis Cedex, France

Correspondence should be addressed to Elías Cueto; ecueto@unizar.es

Received 6 July 2018; Revised 11 September 2018; Accepted 25 September 2018; Published 1 November 2018

Academic Editor: Diyi Chen

Copyright © 2018 Rubén Ibáñez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sparse model identification by means of data is especially cumbersome if the sought dynamics live in a high dimensional space. This usually involves the need for large amount of data, unfeasible in such a high dimensional settings. This well-known phenomenon, coined as the curse of dimensionality, is here overcome by means of the use of separate representations. We present a technique based on the same principles of the Proper Generalized Decomposition that enables the identification of complex laws in the low-data limit. We provide examples on the performance of the technique in up to ten dimensions.

1. Introduction

In recent years there has been a growing interest in incorporating data-driven techniques into the field of mechanics. While almost classical in other domains of science like economics, sociology, etc., big data has arrived with important delay to the field of computational mechanics. It is worth noting that, in our field, the amount of data available is very often no so big, and therefore we speak of data-driven techniques instead of big-data techniques.

Among the first in incorporating data-driven technologies to the field of computational mechanics one can cite the works of Kirchdoerfer et al. [1, 2], or the ones by Brunton et al. [3–5]. Previous attempts exist; however, to construct data-driven identification algorithms, see, for instance [6, 7]. More recently, the issue of compliance with general laws like the ones of thermodynamics has been also achieved, which is a distinct feature of data-driven mechanics [8]. Other applications include the identification of biological systems [9] or financial trading [10], to name but a few.

The problem with high dimensional systems is that data in these systems is often sparse (due precisely to the high

dimensional nature of the phase space) while the system has, on the contrary, low dimensional features—at least very frequently. Based on this, a distinction should be made between methods that require an a priori structure of the sampling points and others which do not require such a regularity.

Regarding the methods that need a rigid structure in the sampling points, the Nonintrusive Sparse Subspace Learning (SSL) method is a novel technique which has proven to be very effective [11]. The basic ingredient behind such a technique is that the parametric space is explored in a hierarchical manner, where sampling points are collocated at the Gauss-Lobato-Chebyshev integration points. Also, using a hierarchical base allows improving the accuracy adding more hierarchical levels without perturbing the previous ones. To achieve such hierarchical property, just the difference at a given point between the real function minus the estimated value, using the precedent hierarchical levels, is propagated. For more details about the method, the reader is referred to [11]. However, in the high-dimensional case, this technique shows severe limitations, as will be detailed hereafter.

On the other hand, nonstructured data-driven techniques are commonly based on Delaunay triangularization techniques, providing an irregular mesh whose nodes coincides with the sampling points. Afterwards, depending on the degree of approximation inside each one of the Delaunay triangles, it gives rise to different interpolation techniques; i.e., linear, nearest, cubic, and natural, among other techniques, are commonly used. Apart from techniques that depend on a given triangularization, it is worth mentioning Kriging interpolants as an appealing technique to provide response surfaces from nonstructured data points. The key ingredient behind such technique is that each sampling point is considered as a realization of a random process. Therefore, defining a spatial correlation function allows to infer the position of unknown points just like providing confidence intervals based on the distance to the measured points. Nevertheless, the calibration of the correlation matrix has an important impact in the performance of the method itself.

Kriging also possesses a very interesting property: it is able to efficiently filter noise and outliers. Therefore, it is expected that it also could help us in problems with noise in the data.

However, in high dimensional settings, all of the just mentioned techniques fail to identify the nature of the system due precisely to the curse of dimensionality. A recent alternative for such a system could be Topological Data Analysis (TDA), which is based on the use of algebraic topology and the concept of *persistent homology* [12]. A sparse version of this technique also exists [13].

Hence, if a competitive data-driven identification technique is desired, such a technique should meet the following requirements:

- (i) *Nonstructured data set*: this characteristic provides versatility to the method. Indeed, when evaluating the response surface requiring a lot of computational effort, recycling previous evaluations of the response surface, which do not coincide with a given structure of the data, may be very useful. In addition, the SSL technique establishes sampling points at locations in the phase space with no physical meaning in an industrial setting.
- (ii) *Robustness with respect to high dimensionality*: triangularization-based techniques suffer when dealing with multidimensional data just because a high dimensional mesh has to be generated. Nevertheless, the separation of variables could be an appealing technique to circumvent the problem of generating such a high dimensional mesh.
- (iii) *Curse of dimensionality*: all previous techniques suffer when dealing with high dimensional data. For instance, the SSL needs 2^D sampling points just to reach the first level of approximation. Thus, when dealing with high dimensional data ($D > 10$ uncorrelated dimensions) plenty of sampling points are required to properly capture a given response surface.

In what follows we present a method based on the concept of separate representations to overcome the curse of

dimensionality. Such separate representation has previously been employed by the authors to construct a priori reduced-order modeling techniques, coined as Proper Generalized Decompositions [14–20]. This will give rise to a sparse Proper Generalized Decomposition (s-PGD in what follows) approach to the problem. We then analyze the just developed technique through a series of numerical experiments in Section 4, showing the performance of the method. Examples in up to ten dimensions are shown. The paper is completed with some discussions.

2. A Sparse PGD (s-PGD) Methodology

2.1. Basics of the Technique. In this section we develop a novel methodology for sparse identification in high dimensional settings. For the ease of the exposition and, above all, representation, but without loss of generality, let us begin by assuming that the unknown objective function $f(x, y)$ lives in \mathbb{R}^2 and that is to be recovered from sparse data. As in previous references, see, for instance [21]; we have chosen to begin with a Galerkin projection, in the form

$$\int_{\Omega} w^*(x, y) (u(x, y) - f(x, y)) dx dy = 0, \quad (1)$$

where $\Omega \subset \mathbb{R}^2$ stands for the—here, still two-dimensional—domain in which the identification is performed and $w^*(x, y) \in \mathcal{C}^0(\Omega)$ is an arbitrary test function. Finally, $u(x, y)$ will be the obtained approximation to $f(x, y)$, still to be constructed. In previous works of the authors [8] as well as in other approaches to the problem (e.g., [21]), this projection is subject to additional constraints of thermodynamic nature. In this work no particular assumption is made in this regard, although additional constraints could be imposed to the minimization problem.

Following the same rationale behind the Proper Generalized Decomposition (PGD), the next step is to express the approximated function $u^M(x, y) \approx u(x, y)$ as a set of separate one-dimensional functions,

$$u^M(x, y) = \sum_{k=1}^M X^k(x) Y^k(y). \quad (2)$$

The determination of the precise form of functional pairs $X^k(x)Y^k(y)$, $k = 1, \dots, M$, is done by first projecting them on a finite element basis and by employing a greedy algorithm such that once the approximation up to order $M - 1$ is known, the new M th order term

$$\begin{aligned} u^M(x, y) &= u^{M-1}(x, y) + X^M(x)Y^M(y) \\ &= \sum_{k=1}^{M-1} X^k(x)Y^k(y) + X^M(x)Y^M(y), \end{aligned} \quad (3)$$

is found by any nonlinear solver (Picard, Newton, . . .).

It is well-known that this approach produces optimal results for elliptic operators (here, note that we have in fact an identity operator acting on u) in two dimensions, see [14] and references therein. There is no proof, however, that

this separate representation will produce optimal results (in other words, will obtain *parsimonious models*) in dimensions higher than two. In two dimensions and with $w^* = u^*$ it provides the singular value decomposition of $f(x, y)$ [15]. Our experience, nevertheless, is that it produces almost optimal results in the vast majority of the problems tested so far.

It is worth noting that the product of the test function $w^*(x, y)$ times the objective function $f(x, y)$ is only evaluated at few locations (the ones corresponding to the experimental measurements) and that, in a general high dimensional setting, we will be in the low-data limit necessarily. Several options can be adopted in this scenario. For instance, the objective function can be first interpolated in the high dimensional space (still 2D in this introductory example) and then integrated together with the test function. Indeed, this will be the so-called *PGD in approximation* [15], commonly used when either $f(x, y)$ is known everywhere and a separated representation is sought or if $f(x, y)$ is known in a separated format but a few pairs M are needed for any reason. Under this rationale the converged solution $u(x, y)$ tries to capture the already interpolated solution in the high dimensional space but in a more compact format. As a consequence, the error due to interpolation of experimental measurements on the high dimensional space will persist in the final separate identified function.

In order to overcome such difficulties, we envisage a projection followed by interpolation method. However since information is just known at P sampling points (x_i, y_i) , $i = 1, \dots, P$, it seems reasonable to express the test function not in a finite element context, but to express it as a set of Dirac delta functions collocated at the sampling points,

$$\begin{aligned} w^*(x, y) &= u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) \\ &= (X^*(x) Y^M(y) + X^M(x) Y^*(y)) \sum_{i=1}^P \delta(x_i, y_i), \end{aligned} \quad (4)$$

giving rise to

$$\begin{aligned} &\int_{\Omega} w^*(x, y) (u(x, y) - f(x, y)) dx dy \\ &= \int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) (u(x, y) - f(x, y)) dx dy \quad (5) \\ &= 0, \end{aligned}$$

The choice of the test function $w^*(x, y)$ in the form dictated by (4) is motivated by the desire of employing a collocation approach while maintaining the symmetry of standard Bubnov-Galerkin projection operation.

2.2. Matrix Form. Let us detail now the finite element projection of the one-dimensional functions $X^k(x)$ and $Y^k(y)$, $k = 1, \dots, M$, (often referred to as *modes*) appearing in (2). Several options can be adopted, ranging from standard piecewise linear shape functions, global nonlinear shape

functions, maximum entropy interpolants, splines, kriging, etc. Regarding the kind of interpolant to use, an analysis will be performed in the sequel. Nevertheless, no matter which precise interpolant is employed, it can be expressed in matrix form as

$$\begin{aligned} X^k(x) &= \sum_{j=1}^N N_j^k(x) \alpha_j^k = [N_1^k(x) \dots N_N^k(x)] \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} \\ &= (\mathbf{N}_x^k)^T \mathbf{a}^k, \end{aligned} \quad (6)$$

$$\begin{aligned} Y^k(y) &= \sum_{j=1}^N N_j^k(y) \beta_j^k = [N_1^k(y) \dots N_N^k(y)] \begin{bmatrix} \beta_1^k \\ \vdots \\ \beta_N^k \end{bmatrix} \\ &= (\mathbf{N}_y^k)^T \mathbf{b}^k, \end{aligned} \quad (7)$$

where α_j^k and β_j^k , $j = 1, \dots, N$, represent the degrees of freedom of the chosen approximation. We employ \mathbf{N}^k as the most usual nomenclature for the shape function vector. It is important to remark that the approximation basis could even change from mode to mode (i.e., for each i). For the sake of simplicity we take the same number of terms for both $X^k(x)$ and $Y^k(y)$, namely, N .

By combining (1), (2), (4), (6), and (7) a nonlinear system of equations is derived, due to products of terms in both spatial directions. An alternate direction scheme is here preferred to linearize the problem, which is also a typical choice in the PGD literature. Note that, when computing modes $X^M(x)$, the variation in the other spatial direction vanishes, $Y^*(y) = 0$, and vice versa.

In order to fully detail the matrix form of the resulting problem, we first employ the notation " \otimes " as the standard tensorial product (i.e., $\mathbf{b} \otimes \mathbf{c} = b_i c_j$) and define the following matrices

$$\begin{aligned} \mathbf{A}_x^{k\ell} &= \mathbf{N}_x^k \otimes \mathbf{N}_x^\ell, \\ \mathbf{A}_y^{k\ell} &= \mathbf{N}_y^k \otimes \mathbf{N}_y^\ell, \\ \mathbf{C}_{xy}^{k\ell} &= \mathbf{N}_x^k \otimes \mathbf{N}_y^\ell. \end{aligned} \quad (8)$$

For the sake of simplicity but without loss of generality, evaluations of the former operators at point (x_i, y_i) are denoted as

$$\begin{aligned} \mathbf{A}_{x_i}^{k\ell} &= \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_x^\ell(x_i), \\ \mathbf{A}_{y_i}^{k\ell} &= \mathbf{N}_y^k(y_i) \otimes \mathbf{N}_y^\ell(y_i), \\ \mathbf{C}_{x_i y_i}^{k\ell} &= \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_y^\ell(y_i). \end{aligned} \quad (9)$$

Equations (10)-(11) below show the discretized version of the terms appearing in the weak form, (1), when computing

modes in the x direction. Again, M stands for the number of modes in the solution $u(x, y)$ while P denotes the number of sampling points.

$$\begin{aligned} & \int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) u(x, y) dx dy \\ &= \sum_{k=1}^M \sum_{i=1}^P \left((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k \right) \left((\mathbf{a}^*)^T \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k \right), \end{aligned} \quad (10)$$

$$\begin{aligned} & \int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) f(x, y) dx dy \\ &= \sum_{i=1}^P f(x_i, y_i) \left((\mathbf{a}^*)^T \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M \right). \end{aligned} \quad (11)$$

Hence, by defining

$$\begin{aligned} \mathbf{M}_x &= \sum_{i=1}^P \left((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{MM} \mathbf{b}^M \right) \mathbf{A}_{x_i}^{MM}, \\ \mathbf{m}_x &= \sum_{k=1}^{M-1} \sum_{i=1}^P \left((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k \right) \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k, \\ \mathbf{f}_x &= \sum_{i=1}^P f(x_i, y_i) \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M, \end{aligned} \quad (12)$$

allows to write a system of algebraic equations

$$\mathbf{M}_x \mathbf{a}^M = \mathbf{f}_x - \mathbf{m}_x. \quad (13)$$

Exactly the same procedure is followed to obtain an algebraic system of equations for \mathbf{b}^M . This allows performing an alternating directions scheme to extract a new couple of $X^M(x)$ and $Y^M(y)$ modes.

This formulation has several aspects that deserve to be highlighted:

- (1) No assumption about $f(x, y)$ has been made other than assuming known its value at sampling points. Indeed, both problems of either interpolating or making a triangulation in a high dimensional space are circumvented due to the separation of variables.
- (2) The operator \mathbf{M}_x is composed of P rank-one updates, meaning that the rank of such operator is at most P . Furthermore, if a subset of measured points share the same coordinate x_i , the entire subset will increase the rank of the operator in one unity.
- (3) The position of the sampling points will constrain the rank of the PGD operators. That is the reason why even if the possibility of having a random sampling of points is available, it is always convenient to perform a smart sampling technique such that the rank in each direction tends to be maximized. Indeed, the higher the rank of the PGD operator is, the more cardinality of \mathbf{a} and \mathbf{b} can be demanded without degenerating into an underdetermined system of equations.

There are plenty of strategies to smartly select the position of the sampling points. They are based on either knowing an a priori error indicator or having a reasonable estimation of the sought response surface. Certainly, an adaptive strategy based on the gradient of the precomputed modes could be envisaged. However, the position of the new sampling points will depend on the response surface calculated using the previous sampling points, making parallelization difficult. That is the reason why Latin hypercube is chosen in the present work. Particularly, Latin hypercube tries to collocate P sampling points in such a way that the projection of those points into x and y axis are as far as possible.

2.3. Choice of the 1D Basis. In the previous section, nothing has been specified about the basis in which each one of the one-dimensional modes was expressed. In this subsection, we will use an interpolant based on Kriging techniques. Simple Kriging has been used throughout history in order to get relatively smooth solutions, avoiding spurious oscillations characteristic of high order polynomial interpolation. This phenomena is called Runge's phenomenon. It appears due to the fact that the sampling point locations are not chosen properly; i.e., they will not be collocated, in general, at the Gauss-Lobato-Chebyshev quadrature points. Kriging interpolants consider each point as a realization of a Gaussian process, so that high oscillations are considered as unlikely events.

Hence, by defining a spatial correlation function based on the relative distance between two points, $\mathcal{D}(x_i - x_j) = \mathcal{D}_{ij}$, an interpolant is created over the separated 1D domain,

$$X^k(x) = \sum_{i=1}^N \frac{\alpha_i^k}{N} + \sum_{j=1}^N \lambda(x - x_j) \left(\alpha_j^k - \sum_{l=1}^N \frac{\alpha_l^k}{N} \right), \quad (14)$$

where $\lambda(x - x_j)$ is a weighting function which strongly depends on the definition of the correlation function and the α_i coefficients are the nodal values associated to the x_i Kriging control points. Note that these control points are not the sampling points. We have chosen this strategy so as to allow us to accomplish an adaptivity strategy that will be described next. In the present work, these control points are uniformly distributed along the 1D domain. Although several definitions of the correlation function exist, a Gaussian distribution is chosen as

$$\mathcal{D}_{ij} = \mathcal{D}(x_i - x_j) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(x_i - x_j)^2 / 2\sigma^2}, \quad (15)$$

where σ is the variance of the Gaussian distribution. Several a priori choices can be adopted to select the value of the variance based on the distance between two consecutive control points, e.g., $\sigma = h \sqrt{(x_{i+1} - x_i)^2}$. The magnitude of h should be adapted depending on the desired global character of the support. To ensure the positivity of the variance, h should be in the interval $]0, +\infty[$.

Let us define now a set of C control points

$$\mathbf{x}^{cp} = [x_1^{cp}, x_2^{cp}, \dots, x_C^{cp}] \quad (16)$$

and the P sampling points

$$\mathbf{x}^{sp} = [x_1^{sp}, x_2^{sp}, \dots, x_P^{sp}]. \quad (17)$$

Let us define in turn a correlation matrix between all control points and a correlation matrix between the control points and the sampling points as

$$\begin{aligned}\mathcal{G}_{ij}^{cp-cp} &= \mathcal{D}(x_i^{cp} - x_j^{cp}), \\ \mathcal{G}_{ij}^{cp-sp} &= \mathcal{D}(x_i^{cp} - x_j^{sp}).\end{aligned}\quad (18)$$

Under these settings, we define a weighting function for each control point and for each sampling point as

$$\Lambda = (\mathcal{G}^{cp-cp})^{-1} \mathcal{G}^{cp-sp}, \quad (19)$$

where $\lambda(x_i^{cp} - x_j^{sp}) = \Lambda_{ij}$.

If we reorganize the terms in the same way that we did in the previous section to have a compact and close format of the shape function \mathbf{N}_x^k , we arrive to

$$\begin{aligned}X^k(x_j^{sp}) &= \sum_{i=1}^N N_i^k(x_j^{sp}) \alpha_i^k \\ &= [N_1^k(x_j^{sp}) \dots N_N^k(x_j^{sp})] \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} \\ &= (\mathbf{N}_{x_j^{sp}}^k)^T \mathbf{a}^k,\end{aligned}\quad (20)$$

where each shape function is given by

$$N_i^k(x_j^{sp}) = \frac{1 - \sum_{j=1}^N \Lambda_{ij}}{N} + \Lambda_{ij}. \quad (21)$$

Figures 1 and 2 depict the appearance of the simple Kriging interpolants using 7 control points uniformly distributed along the domain, for $h = 1$ and $h = 1/3$, respectively. It can be highlighted that both the Kronecker delta (i.e., strict interpolation) and the partition of unity properties are satisfied for any value of h . Moreover, it is worth noting that the higher the variance the correlation function has, the more global the shape functions are. Furthermore, it is known that 99 per cent of the probability of a Gaussian distribution is comprised within an interval of $[m - 3\sigma, m + 3\sigma]$, being m the mean value of the distribution. This issue explains perfectly well why the support of each Gaussian distribution takes 2 elements for the case, where $h = 1/3$. Indeed, the shape of the interpolants is quite similar to standard finite element shape functions, but with a Gaussian profile. The remaining 1 per cent of probability is comprised in the small ridges happening in the middle of the elements.

In light of these results, a family of interpolants based on Kriging can be easily created just selecting the value of the variance within the correlation function. Therefore, globality of the support can be easily adjusted always under the framework of the partition of unity.

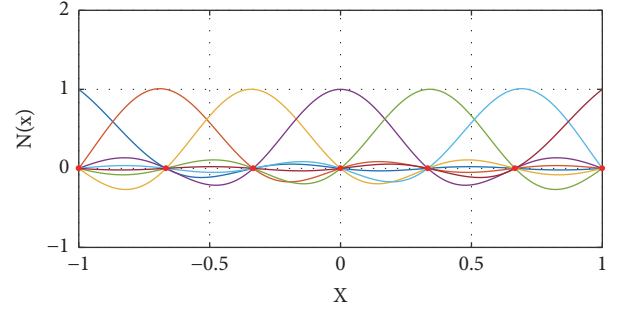


FIGURE 1: Kriging shape functions using $\sigma = \sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

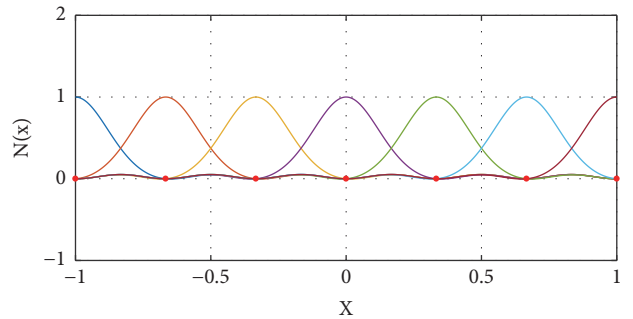


FIGURE 2: Kriging shape functions using $\sigma = (1/3)\sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

2.4. Modal Adaptivity Strategy. In a standard PGD framework, the final solution is approximated as a sum of M modes or functional products; see (2). Each one of the separated modes must be projected onto a chosen basis to render the problem finite dimensional. A standard choice is to select the same basis for each one of the modes:

$$\mathbf{N}^1 = \mathbf{N}^2 = \dots = \mathbf{N}^M. \quad (22)$$

Despite of the fact that this choice seems reasonable, when dealing with nonstructured sparse data, it may not be such. In the previous section we proved that the rank of the separated system strongly depends on the distribution of the data sampling. Therefore, the cardinality of the interpolation basis must not exceed the maximum rank provided by the data sampling. Indeed, this constraint, which provides an upper bound to build the interpolation basis, only guarantees that the minimization is satisfied at the sampling points, without saying anything out of the measured points. Hence, if sampling points are not abundant, in the limit of low-data regime, high oscillations may appear out of these measured points. These oscillations are not desirable since the resulting prediction properties of the proposed method could be potentially decimated.

In order to tackle this problem, we take advantage of the residual-based nature of the PGD. Indeed, the greedy PGD algorithm tries to enrich a solution composed by M modes,

$$u^M(x, y) = \sum_{k=1}^M X^k(x) Y^k(y), \quad (23)$$

just by looking at the residual that accounts for the contribution of the previous modes, as shown in 8).

Therefore, an appealing strategy to minimize spurious oscillations out of the sampling points is to start the PGD algorithm looking for modes with relatively smooth basis (for instance, Kriging interpolants with a few control points). Therefore, an indicator in order to make an online modal adaptive strategy is required. In the present work, we use the norm of the PGD residual,

$$\mathcal{R}_{\mathcal{P}}^M = \frac{1}{\sqrt{P}} \sqrt{\sum_{i \in \mathcal{P}} (f(x_i, y_i) - u^M(x_i, y_i))^2}, \quad (24)$$

where \mathcal{P} is the set of P measured points and $f(x, y)$ is the function to be captured.

In essence, when the residual norm stagnates, a new control mesh is defined, composed by one more control point and always uniformly distributed, following

$$\Delta \mathcal{R}_{\mathcal{P}}^M = \mathcal{R}_{\mathcal{P}}^M - \mathcal{R}_{\mathcal{P}}^{M-1} < \epsilon_r. \quad (25)$$

By doing this, oscillations are reduced, since higher-order basis will try to capture only what remains in the residual. Here, ϵ_r is a tolerance defining the resilience of the sPGD to increase the cardinality of the interpolation basis. The lower ϵ_r is, the more resilient method is to increase the cardinality.

To better understand the method, we will quantify the error for two set of points: the first set is associated with the sampling points, \mathcal{P} ,

$$\mathcal{E}_{\mathcal{P}} = \frac{1}{\#\mathcal{P}} \sum_{s \in \mathcal{P}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}}, \quad (26)$$

where $f(x_s, y_s)$ is assumed not to vanish and where \mathcal{L} also includes points other than the sampling points. This is done in order to validate the algorithm, by evaluating the reference solution—which is a priori unknown in a general setting—at points different to the sampling ones,

$$\mathcal{E}_{\mathcal{L}} = \frac{1}{\#\mathcal{L}} \sum_{s \in \mathcal{L}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}}. \quad (27)$$

Since the s-PGD algorithm minimizes the error only at the sampling points \mathcal{P} it is reasonable to expect that $\mathcal{E}_{\mathcal{P}} \leq \mathcal{E}_{\mathcal{L}}$.

2.5. A Preliminary Example. To test the convergence of the just presented algorithm, we consider

$$f_1(x, y) = (\cos(3\pi x) + \sin(3\pi y))y^2 + 4, \quad (28)$$

which presents a quite oscillating behavior along the x direction, whereas the y direction is quadratic. We are interested in capturing such a function in the domain $\Omega_y = \Omega_x = [-1, 1]$.

Figures 3 and 4 show the errors $\mathcal{E}_{\mathcal{P}}$ and $\mathcal{E}_{\mathcal{L}}$ in identifying the function $f_1(x, y)$. In this case, we consider two distinct

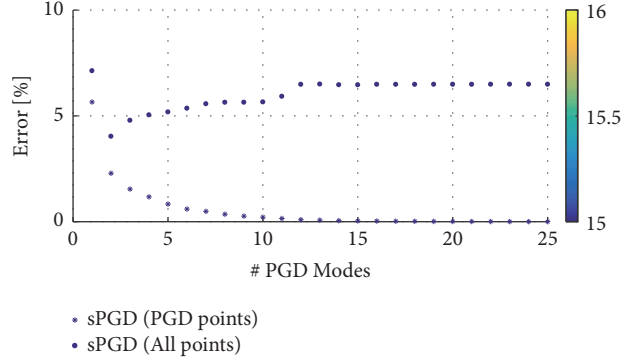


FIGURE 3: $\mathcal{E}_{\mathcal{P}}$ (points) and $\mathcal{E}_{\mathcal{L}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. No modal adaptivity.

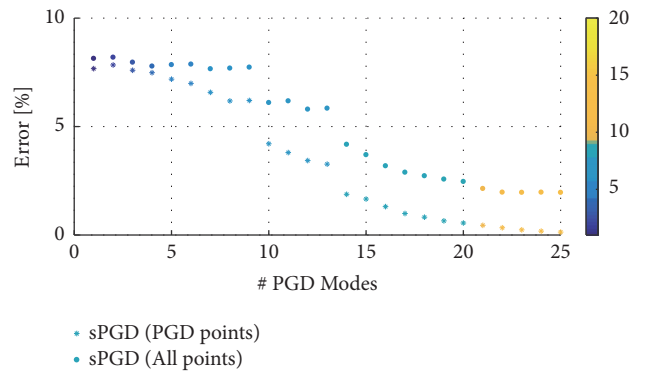


FIGURE 4: $\mathcal{E}_{\mathcal{P}}$ (points) and $\mathcal{E}_{\mathcal{L}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. Modal adaptivity based on the residual, $\epsilon_r = 1e-2$.

possibilities: no modal adaptivity at all and a modal adaptivity based on the residual, respectively. Several aspects can be highlighted. The first one is that $\mathcal{E}_{\mathcal{L}}$ (asterisks) decreases much faster when there is no modal adaptivity. This is expected, since we are minimizing with a richer basis since the very beginning, instead of starting with smooth functions like in the residual based approach. However, even if the minimization in the sampling points is well achieved, when no modal adaptivity is considered, the error out of the sampling points may increase as the solution is enriched with new modes. Nevertheless, the residual-based modal adaptivity alleviates this problem. As it can be noticed, starting with relatively smooth functions drives the solution out of the sampling points to be smooth as well, avoiding the problem of high oscillations appearing out of the sampling points.

3. A Local Approach to s-PGD

It is well-known that, as in POD, reduced basis or, in general, any other linear model reduction technique, PGD gives poor results—in the form of a nonparsimonious prediction—when the solution of the problem lives in a highly nonlinear manifold. Previous approaches to this difficulty included the employ of nonlinear dimensionality reduction techniques such as Locally Linear Embeddings [22], kernel Principal

Component Analysis [23, 24] or isomap techniques [25]. Another, distinct, possibility, is to employ a local version of PGD [18], in which the domain is sliced so that at every sub-region PGD provides optimal or nearly optimal results. We explore this last option here for the purpose of sparse regression, although a bit modified, as will be detailed hereafter.

The approach followed herein is based on the employ of the partition of unity property [26, 27]. In essence, it is well-known that any approximating function (like finite element shape functions, for instance) that forms a partition of unity can be enriched with an arbitrary function such that the resulting approximation inherits the smoothness of the partition of unity and the approximation properties of the enriching function.

With this philosophy in mind, we proposed to enrich a finite element mesh with an s-PGD approximation. The resulting approximation will be local, due to the compact support of finite element approximation, while inheriting the good approximation properties, already demonstrated, of s-PGD. In essence, what we propose is to construct an approximation of the type

$$u(x, y) \approx \sum_{i \in \mathcal{I}} N_i(x, y) u_i + \sum_{p \in \mathcal{I}_{\text{enr}}} \sum_{e \in \mathcal{I}_p} N_e(x, y) \underbrace{\sum_{k=1}^M X_p^k(x) Y_p^k(y)}_{f_p^{\text{enr}}(x, y)}, \quad (29)$$

where \mathcal{I} represents the node set in the finite element mesh, \mathcal{I}_{enr} the set of enriched nodes, u_i are the nodal degrees of freedom of the mesh, \mathcal{I}_p is the number of finite elements covered by node p shape function's support and $X_p^k(x)$ and $Y_p^k(y)$ functions are the k -th one-dimensional PGD modes enriching node p , that in fact constitute an enriching function $f_p^{\text{enr}}(x, y)$.

Of course, as already introduced in Eqs. (6) and (7), every PGD mode is in turn approximated by Galerkin projection on a judiciously chosen basis. In other words,

$$u(x, y) \approx \sum_{i \in \mathcal{I}} N_i(x, y) u_i + \sum_{p \in \mathcal{I}_{\text{enr}}} \sum_{e \in \mathcal{I}_p} N_e(x, y) \sum_{k=1}^M (\mathbf{N}_x^k)^T \mathbf{a}_p^k (\mathbf{N}_y^k)^T \mathbf{b}_p^k, \quad (30)$$

with \mathbf{a}_p^k and \mathbf{b}_p^k the nodal values describing each one-dimensional PGD mode.

In this framework, the definition of a suitable test function can be done in several ways. As a matter of fact, the test function can be expressed as the sum of a finite element and a PGD contribution,

$$u^* = u_{\text{FEM}}^* + u_{\text{PGD}}^*, \quad (31)$$

so that an approach similar to that of Eq. (4) can be accomplished.

An example of the performance of this approach is included in Section 4.4.

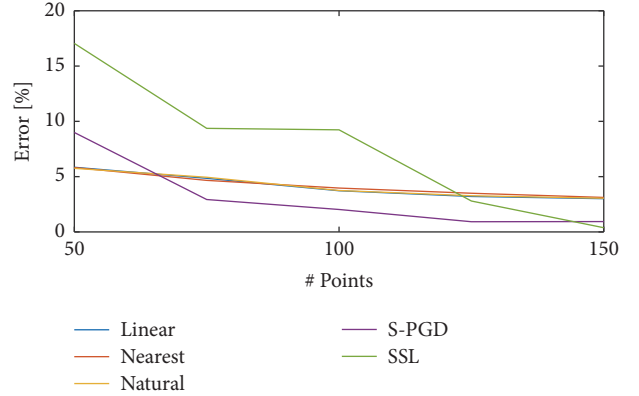


FIGURE 5: $\mathcal{E}_{\mathcal{I}}$ of $f_1(x, y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

4. Numerical Results

The aim of this section is to compare the ability of sparse model identification for different interpolation techniques. On one hand, the performance of standard techniques based on Delaunay triangulation such as linear, nearest neighbor or cubic interpolation is compared. Even though these techniques are simple, they allow to have a nonstructured sampling point set since they rely on a Delaunay triangulation. On the other hand, the results are compared to the Sparse Subspace Learning (SSL) [11]. The convergence and robustness of this method is proven to be very effective since the points are collocated at the Gauss-Lobato-Chebyshev points. However, two main drawbacks appear considering this method. The first one is that there is a high concentration of points in the boundary of the domain, so that this quadrature is meant for functions that vary mainly along the boundary. Indeed, if the variation of the function appears in the middle of the domain, many sampling points will be required to converge to the exact function. The second one is that the sampling points have to be located at specific points in the domain. The s-PGD method using simple Kriging interpolants will be compared as well.

The numerical results are structured as follows: first two synthetic 2D functions are analyzed; secondly, two 2D response surfaces coming from a thermal problem and a Plastic Yield function are reconstructed; finally, a 10D synthetic function is reconstructed by means of the s-PGD algorithm.

4.1. 2D Synthetic Functions. The first considered function is $f_1(x, y)$, as introduced in the previous section. Figure 5 shows the reconstruction error ($\mathcal{E}_{\mathcal{I}}$) of $f_1(x, y)$ for different sampling points. As it can be noticed, the s-PGD algorithm performs well for a wide range of sampling points. Nevertheless, the SSL method is the one presenting the lower error level when there are more than 150 sampling points.

A second synthetic function is defined as

$$f_2(x, y) = \cos(3xy) + \log(x + y + 2.05) + 5. \quad (32)$$

This function is intended to be reconstructed in the domain $\Omega_x = \Omega_y = [-1, 1]$. It was chosen in such a way that it is

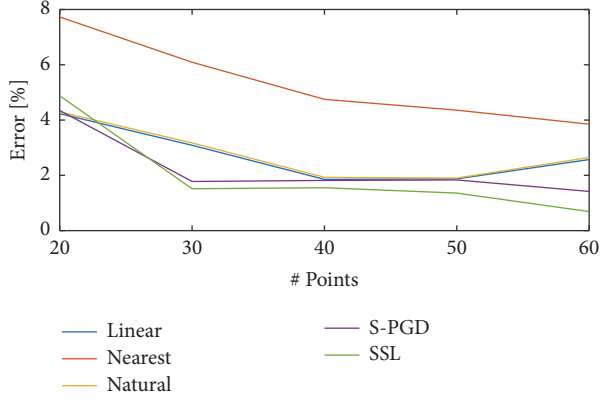


FIGURE 6: $\mathcal{E}_{\mathcal{P}}$ of $f_2(x, y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

relatively smooth in the center of the domain, whereas the main variation is located along the boundary of the domain. Indeed, this function is meant to show the potential of the SSL technique.

Figure 6 shows the reconstruction error of the $f_2(x, y)$ function for different interpolation techniques. As it can be noticed, both SSL and s-PGD methods are the ones that present the best convergence properties. If the number of points is increased even more, the SSL method is the one that presents the lowest interpolation error. They are followed by linear and natural neighbor interpolations. Finally, the nearest neighbor method is the one presenting the worst error for this particular case.

4.2. 2D Response Surfaces Coming from Physical Problems.

Once the convergence of the methods have been unveiled for synthetic functions, it is very interesting to analyze the power of the former methods by trying to identify functions that are coming from either simulations or models popular in the computational mechanics community. Indeed, two functions will be analyzed: the first one is an anisotropic Plastic Yield function, whereas the second one is a solution coming from a quasistatic thermal problem with varying source term and conductivity.

Figure 7 shows the Yld2004-18p anisotropic plastic yield function, defined by Barlat et al. in [28]. Under plane stress hypothesis, this plastic yield function is a convex and closed surface defined in a three-dimensional space. Therefore, the position vector of an arbitrary point in the surface can be easily parameterized in cylindrical coordinates as $R(\theta, \sigma_{xy})$. The $R(\theta, \sigma_{xy})$ function for the Yld2004-18p is shown in Figure 8, where anisotropies can be easily seen. Otherwise, the radius function will be constant for a given σ_{xy} .

Figure 9 shows the error in the identification of Barlat's plastic yield function Yld2004-18p. As it can be noticed, the s-PGD technique outperforms the rest of techniques. Indeed, the s-PGD is exploiting the fact that the response surface is highly separable.

As mentioned above, the second problem is the sparse identification of the solution of a quasistatic thermal problem modeled by

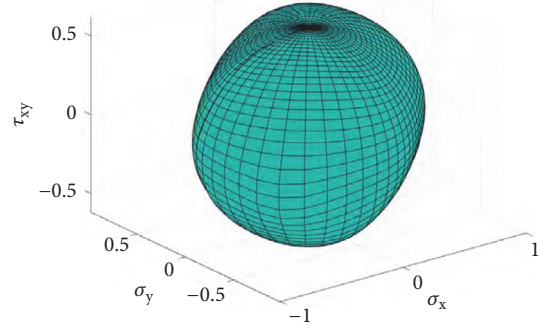


FIGURE 7: Barlat's Yld2004-18p function under plane stress hypothesis.

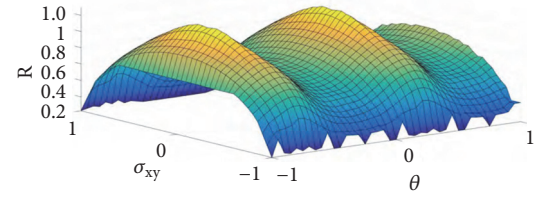


FIGURE 8: $R(\theta, \sigma_{xy})$ function for Barlat's Yld2004-18p yield function.

$$\begin{aligned} \nabla \cdot (\eta(x, t) \nabla (u(x, t))) &= f(t), \\ \text{in } \Omega_x \times \Omega_t &= [-1, 1] \times [-1, 1], \end{aligned} \quad (33)$$

where conductivity varies in space-time as

$$\eta(x, t) = (1 + 10 \text{abs}(x) + 10x^2) \log(t + 2.5) \quad (34)$$

$$u(1, t) = 2$$

$$f(x, t) = 10 \cos(3\pi t) \quad (35)$$

$$u(-1, t) = 2,$$

and the source term varies in time. Homogeneous Dirichlet boundary conditions are imposed at both spatial boundaries and no initial conditions are required due to quasistationarity assumptions.

Figure 10 shows the evolution of the temperature field as a function of space time for the set of (33)-(35). It can be noticed how the variation of the temperature throughout time is caused mainly due to the source term. However, conductivity modifies locally the curvature of the temperature along the spatial axis. Symmetry with respect the $x = 0$ axis is preserved due to the fact that the conductivity presents a symmetry along the same axis.

Figure 11 shows the performance of each one of the techniques when trying to reconstruct the temperature field from certain sampling points. As can be noticed, the s-PGD in conjunction with Kriging interpolants is the one that presents the fastest convergence rate to the actual function, which is considered unknown. It is followed by linear and natural interpolations. The SSL method presents a slow convergence rate in this case, due to the fact that the main variation of the function $u(x, t)$ is happening in the center of the domain and not in the boundary.

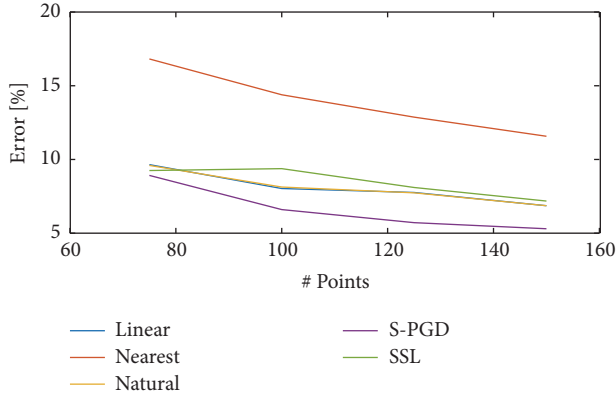


FIGURE 9: $\mathcal{E}_{\mathcal{P}}$ of $R(\theta, \sigma_{xy})$ varying $\#\mathcal{P}$ for different sparse identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 5 \cdot 10^{-4}$.

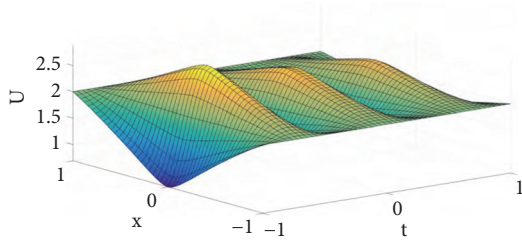


FIGURE 10: Quasistatic solution to the thermal problem $u(x, t)$.

4.3. A 10D Multivariate Case. In this subsection, we would like to show the scalability that s-PGD presents when dealing with relatively high-dimensional spaces. Since our solution is expressed in a separated format, an N dimensional problem (ND) is solved as a sequence of N 1D problems, which are solved using a fixed-point algorithm in order to circumvent the nonlinearity of the separation of variables.

The objective function that we have used to analyze the properties of the s-PGD is defined as

$$f_3(x_1, x_2, \dots, x_N) = 2 + \frac{1}{8} \sum_{i=1}^N x_i + \prod_{i=1}^N x_i + \prod_{i=1}^N x_i^2, \quad (36)$$

with $N = 10$ in this case.

Figure 12 shows the error convergence in both sampling points ($\mathcal{E}_{\mathcal{P}}$, asterisks) and points out of the sampling ($\mathcal{E}_{\mathcal{L}}$, filled points). The \mathcal{L} data set was composed by 3000 points, and the \mathcal{P} data subset for the s-PGD algorithm was composed by 500 points. The number of points required to properly capture the hypersurface has increased with respect to the 2D examples due to the high dimensionality of the problem. Special attention has to be paid when increasing the cardinality of the interpolant basis without many sampling points, because the problem of high oscillations outside the control points may be accentuated.

4.4. An Example of the Performance of the Local s-PGD. The last example corresponds to the sparse regression of an intricate surface that has been created by mixing three different Gaussian surfaces so as to generate a surface with

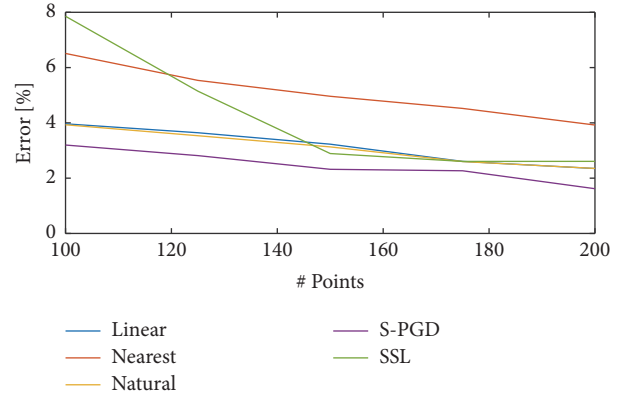


FIGURE 11: $\mathcal{E}_{\mathcal{P}}$ of $u(x, t)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 2.5 \cdot 10^{-3}$.

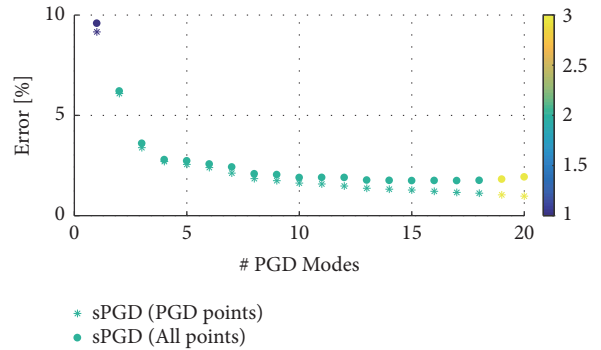


FIGURE 12: $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_3(x_1, x_2, \dots, x_N)$, $\#\mathcal{P} = 500$, $\#\mathcal{L} = 3000$. Modal adaptivity based on the residual, $\epsilon_r = 1e - 3$.

no easy separate representation (a nonparsimonious model, if we employ a different vocabulary). The appearance of this synthetic surface is shown in Figure 13.

The sought surface is defined in the domain $\Omega = [0, 1]^2$, which has been split into the finite element mesh shown in Figure 14. Every element in the mesh has been colored according to the number of enriching PGD functions, ranging from a single one to four. The convergence plot of this example as a function of the number of PGD modes added to the approximating space is included in Figure 15.

5. Conclusions

In this paper we have developed a data-based sparse reduced-order regression technique under the Proper Generalized Decomposition framework. This algorithm combines the robustness typical of the separation of variables together with properties of collocation methods in order to provide with parsimonious models for the data at hand. The performance of simple Kriging interpolation has proven to be effective when the sought model presents some regularity. Furthermore, a modal adaptivity technique has been proposed in order to avoid high oscillations out of the sampling points, characteristic of high-order interpolation methods when data is sparse.

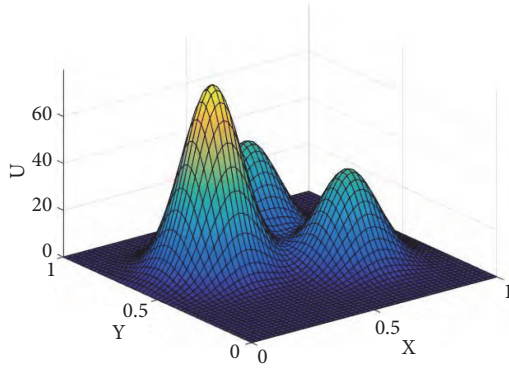


FIGURE 13: A synthetic surface generated by superposition of three different Gaussians, that is to be approximated by local s-PGD techniques.

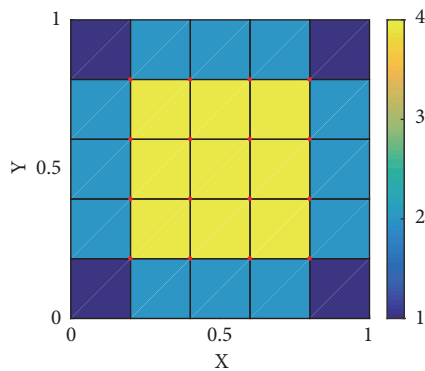


FIGURE 14: Finite element mesh for the example in Section 4.4. All the internal nodes have been enriched.

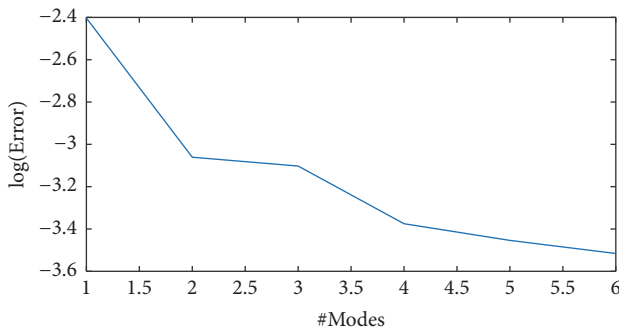


FIGURE 15: Convergence plot for the example in Section 4.4.

For problems in which the result lives in a highly nonlinear manifold, a local version of the technique, which makes use of the partition of unity property, has also been developed. This local version outperforms the standard one for very intricate responses.

The s-PGD method has been compared advantageously versus other existing methods for different example functions. Finally, the convergence of s-PGD method for a high dimensional function has been demonstrated as well.

Although the sparsity of the obtained solution could not seem evident for the reader, we must highlight the fact

that the very nature of the PGD strategy *a priori* selects those terms on the basis that it plays a relevant role in the approximation. So to speak, PGD algorithms automatically discard those terms that in other circumstances will be weighted by zero. Sparsity, in this sense, is equivalent in this context to the number of sums in the PGD separated approximation. If only a few terms are enough to reconstruct the data—as is almost always the case—then sparsity is guaranteed in practice.

Sampling strategies other than the Latin hypercube method could be examined as well. This and the coupling with error indicators to establish good stopping criteria constitute our effort of research at this moment. In fact, the use of reliable error estimators could even allow for the obtention of adaptive samplings in which the cardinality of the basis could be different along different directions.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant agreement no. 675919, also the Spanish Ministry of Economy and Competitiveness through Grants nos. DPI2017-85139-C2-1-R and DPI2015-72365-EXP, and the Regional Government of Aragon and the European Social Fund, research group T24 17R.

References

- [1] T. Kirchdoerfer and M. Ortiz, "Data-driven computational mechanics," *Computer Methods Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016.
- [2] T. Kirchdoerfer and M. Ortiz, "Data driven computing with noisy material data sets," *Computer Methods Applied Mechanics and Engineering*, vol. 326, pp. 622–641, 2017.
- [3] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [4] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (sindy)," <https://arxiv.org/abs/1605.06682>.
- [5] M. Quade, M. Abel, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for rapid model recovery," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 6, 063116, 10 pages, 2018.
- [6] D. Gonzalez, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, and F. Chinesta, "Proper generalized decomposition based dynamic

- data driven inverse identification,” *Mathematics and Computers in Simulation*, vol. 82, no. 9, pp. 1677–1695, 2012.
- [7] B. Peherstorfer and K. Willcox, “Data-driven operator inference for nonintrusive projection-based model reduction,” *Computer Methods Applied Mechanics and Engineering*, vol. 306, pp. 196–215, 2016.
- [8] D. González, F. Chinesta, and E. Cueto, “Thermodynamically consistent data-driven computational mechanics,” *Continuum Mechanics and Thermodynamics*, 2018.
- [9] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Inferring Biological Networks by Sparse Identification of Non-linear Dynamics,” *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, 2016.
- [10] J. Mann and J. N. Kutz, “Dynamic mode decomposition for financial trading strategies,” *Quantitative Finance*, vol. 16, no. 11, pp. 1643–1655, 2016.
- [11] D. Borzacchiello, J. V. Aguado, and F. Chinesta, “Non-intrusive Sparse Subspace Learning for Parametrized Problems,” *Archives of Computational Methods in Engineering: State-of-the-Art Reviews*, pp. 1–24, 2017.
- [12] C. Epstein, G. Carlsson, and H. Edelsbrunner, “Topological data analysis,” *Inverse Problems*, vol. 27, no. 12, 2011.
- [13] W. Guo, K. Manohar, S. L. Brunton, and A. G. Banerjee, “Sparse-TDA: Sparse Realization of Topological Data Analysis for Multi-Way Classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1403–1408, 2018.
- [14] F. Chinesta, A. Ammar, and E. Cueto, “Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models,” *Archives of Computational Methods in Engineering: State-of-the-Art Reviews*, vol. 17, no. 4, pp. 327–350, 2010.
- [15] F. Chinesta, R. Keunings, and A. Leygue, *The proper generalized decomposition for advanced numerical simulations*, Springer-Briefs in Applied Sciences and Technology, Springer, Cham, 2014.
- [16] F. Chinesta and P. Ladeveze, Eds., *Separated representations and PGD-based model reduction*, vol. 554 of *CISM International Centre for Mechanical Sciences. Courses and Lectures*, Springer, 2014.
- [17] D. Gonzalez, A. Ammar, F. Chinesta, and E. Cueto, “Recent advances on the use of separated representations,” *International Journal for Numerical Methods in Engineering*, vol. 81, no. 5, pp. 637–659, 2010.
- [18] A. Badas, D. González, I. Alfaro, F. Chinesta, and E. Cueto, “Local proper generalized decomposition,” *International Journal for Numerical Methods in Engineering*, vol. 112, no. 12, pp. 1715–1732, 2017.
- [19] E. Cueto, D. González, and I. Alfaro, *Proper generalized decompositions*, Springer Briefs in Applied Sciences and Technology, Springer, 2016.
- [20] D. González, A. Badas, I. Alfaro, F. Chinesta, and E. Cueto, “Model order reduction for real-time data assimilation through extended Kalman filters,” *Computer Methods Applied Mechanics and Engineering*, vol. 326, pp. 679–693, 2017.
- [21] J.-C. Loiseau and S. L. Brunton, “Constrained sparse Galerkin regression,” *Journal of Fluid Mechanics*, vol. 838, pp. 42–67, 2018.
- [22] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [23] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [24] B. Schölkopf, A. Smola, and K. R. Müller, “Kernel principal component analysis,” in *Advances in kernel methods - support vector learning*, pp. 327–352, MIT Press, 1999.
- [25] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [26] J. M. Melenk and I. Babuška, “The partition of unity finite element method: basic theory and applications,” *Computer Methods Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 289–314, 1996.
- [27] I. Babuška and J. M. Melenk, “The partition of unity method,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 4, pp. 727–758, 1997.
- [28] J. W. Yoon, F. Barlat, R. E. Dick, and M. E. Karabin, “Prediction of six or eight ears in a drawn cup based on a new anisotropic yield function,” *International Journal of Plasticity*, vol. 22, no. 1, pp. 174–193, 2006.

Research Article

A Conceptual Approach to Complex Model Management with Generalized Modelling Patterns and Evolutionary Identification

Sergey V. Kovalchuk ¹, Oleg G. Metsker,¹ Anastasia A. Funkner ¹,
Ilya O. Kisliakovskii ¹, Nikolay O. Nikitin,¹ Anna V. Kalyuzhnaya ¹,
Danila A. Vaganov,^{1,2} and Klavdiya O. Bochenina ¹

¹ITMO University, Saint Petersburg, Saint Petersburg, Russia

²University of Amsterdam, Amsterdam, Netherlands

Correspondence should be addressed to Sergey V. Kovalchuk; sergey.v.kovalchuk@gmail.com

Received 1 June 2018; Accepted 17 September 2018; Published 1 November 2018

Guest Editor: Rafael Gómez-Bombarelli

Copyright © 2018 Sergey V. Kovalchuk et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Complex systems' modeling and simulation are powerful ways to investigate a multitude of natural phenomena providing extended knowledge on their structure and behavior. However, enhanced modeling and simulation require integration of various data and knowledge sources, models of various kinds (data-driven models, numerical models, simulation models, etc.), and intelligent components in one composite solution. Growing complexity of such composite model leads to the need of specific approaches for management of such model. This need extends where the model itself becomes a complex system. One of the important aspects of complex model management is dealing with the uncertainty of various kinds (context, parametric, structural, and input/output) to control the model. In the situation where a system being modeled, or modeling requirements change over time, specific methods and tools are needed to make modeling and application procedures (metamodeling operations) in an automatic manner. To support automatic building and management of complex models we propose a general evolutionary computation approach which enables managing of complexity and uncertainty of various kinds. The approach is based on an evolutionary investigation of model phase space to identify the best model's structure and parameters. Examples of different areas (healthcare, hydrometeorology, and social network analysis) were elaborated with the proposed approach and solutions.

1. Introduction

Today the area of modeling and simulation of complex systems evolves rapidly. A complex system [1] is usually characterized by a large number of elements, complex long-distance interaction between elements, and multiscale variety. One of the results of the area's development is growing complexity of the models used for investigation of complex systems. As a result, contemporary model of a complex system could be easily characterized by the same features as a natural complex system. Usually, a complexity of a model is considered in tight relation to a complexity of a modeling system. Nevertheless, in many cases, the complexity of a model does not mimic the complexity of a system under investigation (at least exactly). It leads to additional issues in managing a complex model during identification, calibration, data assimilation,

verification, validation, and application. One of the core reasons for these issues is the uncertainty of various kinds [2, 3] applied on levels of system, data, and model. In addition, complexity is even more extended within multidisciplinary models and models which incorporate additional complex or/and third-party submodels. From the application point of view, complex models are often difficult to support and integrate with a practical solution because of a low level of automation and high modeling skills needed to support and adapt a model to the changing conditions.

On the other hand, recently evolutionary approaches are popular for solving various types of model-centered operations like model identification [4], equation-free methods [5], ensemble management [6], data assimilation [7], and others. Evolutionary computation (EC) provides the ability to implement automatic optimization and dynamic adaptation

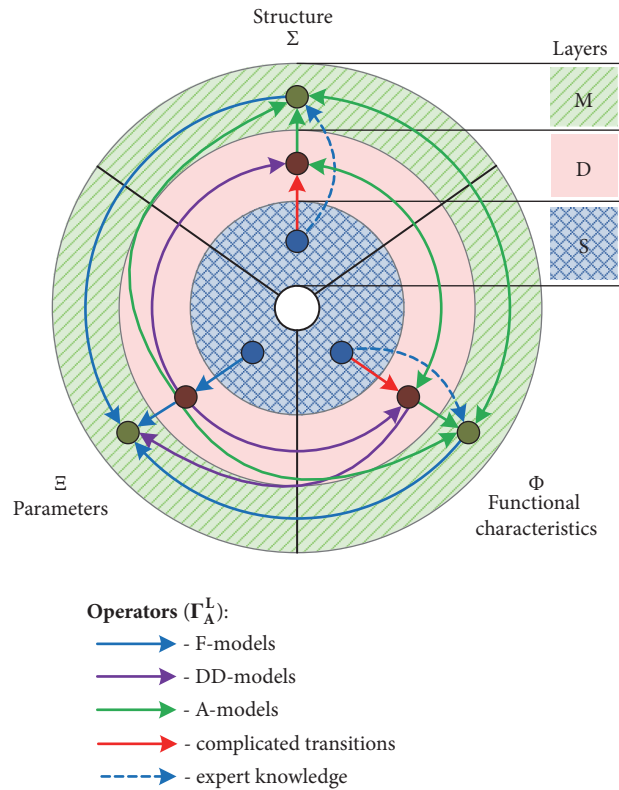


FIGURE 1: Basic concepts of complex modeling on model (M), data (D), and system (S) layers.

of the system within a complex state space. Still, most of the solutions are still tightly related to the application and modeling system.

Within the current research, we are trying to develop a unified conceptual and technological approach to support core operation with a complex model by distinguishing concepts and operations on model, data, and system levels. We consider a combination of EC and data-driven approaches as a tool for building intelligent solutions for more precise and systematic managing (and lowering) uncertainty and providing the required level of automation, adaptability, and extendibility.

2. Conceptual Basis

The proposed approach is based on several key ideas, aimed to extend uncertainty management in complex system modeling and simulation.

(1) Disjoint consideration of model, data, and system in terms of structure, behavior, and quality is aimed toward a system-level review of modeling and simulation process and distinguishes between the uncertainties of various kinds originated from different level [8].

(2) Intelligent technologies like data mining, process mining, machine learning, and knowledge-based approaches are to be hired to fill the gap in automation of modeling and simulation. Key sources for the development of such solution include formalization of various knowledge within composite

solutions [9] and data-driven technologies to support the identification of model components.

(3) EC approaches are widespread in modeling and simulation of complex systems [8, 10]. We believe that systematization of this process with separate consideration of spaces for a system (with its subsystems) and a model (with its submodels) could enhance such solutions significantly.

(4) The aim of the approach's development is twofold. First, it is aimed towards automation of modeling operations to extend the functionality of possible model-based applications. Second, working with a combination of EC and intelligent data-driven technologies could be considered as an additional knowledge source for system and model analysis.

Furtherly, this section considers the conceptual basis of the proposed approach with a special focus on the role of EC algorithms and data-driven intelligent technologies for building and exploiting complex models.

2.1. Core Concepts. To distinguish between main modeling concepts and operations, we propose a conceptual framework (see Figure 1) for consideration of key processes and operations during modeling of the complex system. The framework may be considered as a generalization and extension of a framework [11, 12] previously defined and used by authors for ensemble-based simulation. Current research extends the concept beyond ensemble-based simulation. It is mainly focused on complex modeling in general with identification of key model management procedures and important

artifacts which can be used for model development and application.

The proposed framework considers three main layers of complex systems' modeling, namely, model (M), data (D), and system (S). Main operations (arrows on the diagram) within the framework are defined within three concepts: quantitative parameters (Ξ), functional characteristics (Φ), and structure (Σ). We denote operations by Γ_L^A , where A and L stay for concepts and layer (respectively) involved in the operation. Transitions between concepts and between layers are denoted by $A_1 \rightarrow A_2$ and $L_1 \rightarrow L_2$ respectively; e.g., operator $\Gamma_{S \rightarrow D}^\Xi$ reflects observation of quantitative parameters and operator $\Gamma_{D \rightarrow M}^\Xi$ stays for basic data assimilation. Also, a set of operators may refer to a single modeling operation; e.g., operators $\Gamma_M^{\Phi \rightarrow \Xi}$ and $\Gamma_M^{\Sigma \rightarrow \Xi}$ are often implemented within a single monolithic model. Mainly, operators are related to the specific submodel within a complex model. We consider three key classes of models. F-models are usually classical continuous models developed with knowledge of a system. DD-models are data-driven models based on analysis of available data sets with corresponding techniques (statistics, data mining, process mining, etc.). A-models are mainly intelligent components of a system usually based on machine learning or knowledge-based approaches. Also, we consider EC-based components as belonging to A-models class.

A key problem within complex system modeling and simulation is related to the absent or at least significantly limited possibility to observe the structure and functional characteristics of the system (operators $\Gamma_{S \rightarrow D}^\Phi$ and $\Gamma_{S \rightarrow D}^\Sigma$) directly. The general solution usually includes implicit substitution of the operators with the expertise of modeler (operators $\Gamma_{S \rightarrow M}^\Phi$ and $\Gamma_{S \rightarrow M}^\Sigma$). Still, the more complex the system under investigation and the model are, the more limited those operations are. To overcome this issue, additional DD-models are involved (operators $\Gamma_D^{\Xi \rightarrow \Sigma}$ and $\Gamma_D^{\Xi \rightarrow \Phi}$ for mining in available data, $\Gamma_{D \rightarrow M}^{\Phi \rightarrow \Xi}$ for extended discovery of model parameters for various functional characteristics). Also, A-models are hired to extend expert knowledge in discovery of M -layer concepts with either formalized knowledge or knowledge discovered in data with machine learning approaches (operators $\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Sigma}$, $\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Phi}$, $\Gamma_{D \rightarrow M}^\Sigma$, and $\Gamma_{D \rightarrow M}^\Phi$ for direct discovery of structure and functional characteristics directly and operators $\Gamma_D^{\Sigma \rightarrow \Phi}$, $\Gamma_D^{\Phi \rightarrow \Sigma}$, $\Gamma_M^{\Sigma \rightarrow \Phi}$, and $\Gamma_M^{\Phi \rightarrow \Sigma}$ for interconnection of discovered characteristics in available data and within the used model). In the proposed approach, primary attention is paid to these kinds of solutions where DD- and A-models enable enhancement of complex modeling process with an additional level of automation, adaptation, and knowledge providing.

2.2. Complex Modeling Patterns. Considering the defined conceptual framework, we identify several patterns of modeling and simulation of a complex system (see Figure 2). The patterns are defined as combinations in a context of the framework described previously (3 layers, 3 concepts). An essential idea of the proposed patterns is systematization of

complex model management approaches with combinations of expertise, intelligent solution (A-models), DD-models, and EC.

The pattern extends the operators described in Section 2.1 for model building with operators for model application (i.e., modelling and simulation) and results analysis (e.g., assessing model quality) required for automated model identification. These additional Operators are denoted with Γ_L^A and similar notation for indices.

P1. Regular modeling of a system (Figure 2(a)) is a basic pattern usually applied to discover new knowledge on the system under investigation. A model is built using (a) expertise of modeled for identification of structure and functional characteristics of the model ($\Gamma_{S \rightarrow M}^\Phi$ and $\Gamma_{S \rightarrow M}^\Sigma$); (b) available input data usually representing quantitative parameters of a system considered as a static input of the model or source for data assimilation (DA) via operators $\Gamma_{S \rightarrow D}^\Xi$ and $\Gamma_{D \rightarrow M}^\Xi$. Results of model application ($\Gamma_{M \rightarrow D}^\Xi$, $\Gamma_{M \rightarrow D}^\Sigma$, and $\Gamma_{M \rightarrow D}^\Phi$) could be considered from descriptive (mainly structural or quantitative characteristics) or predictive (often forecasting or other functional characteristics). The obtained results are analyzed in comparison to available information about the investigated system ($\Gamma_{D \rightarrow S}^\Xi$, $\Gamma_{D \rightarrow S}^\Sigma$, and $\Gamma_{D \rightarrow S}^\Phi$) forming an optimization loop which can be considered within the scope of all three concepts. Certain limitations within this pattern being applied to complex system modeling and simulation are introduced by two factors. First, working with complex structural and functional characteristics of the model requires a high level of expertise which leads to a limitation of extensibility and automation of model operation. Second, performing optimization in a loop with most algorithms require multiple runs of a model. As a result, computational-intensive models have limitations in optimization-based operations (identification, calibration, etc.) due to performance reasons.

P2. Data-driven modeling (Figure 2(b)) provides an extension to the modeling operation describing the relationship between data attributes. Application of data-driven models may be considered as replacement of actual "full" model, providing (a) information about structure of system and model with data mining (DM) and process mining (PM) techniques ($\Gamma_D^{\Xi \rightarrow \Sigma}$); (b) generating surrogate models for functional characteristics ($\Gamma_D^{\Xi \rightarrow \Phi}$); (c) providing estimation of investigated parameters with machine learning (ML) algorithms and models ($\Gamma_D^{\Xi \rightarrow \Xi}$). In contrast to the previous pattern data-driven models usually operate quickly (although it could require significant time to train the model). Still, such models have lower quality than original "full" models. Nevertheless, combining this pattern with others provides significant enhancement in functionality and performance; e.g., data-driven models can be used in optimization loop (see previous pattern).

P3. Ensemble-based modeling (Figure 2(c)) extends P1 for working with sets of objects (models, data sets, and states) reflecting uncertainty, variability, or alternative solutions (e.g., models). Previously [11] we identified 5 classes of ensembles (see E1-E5 in Figure 2(c)): decomposition

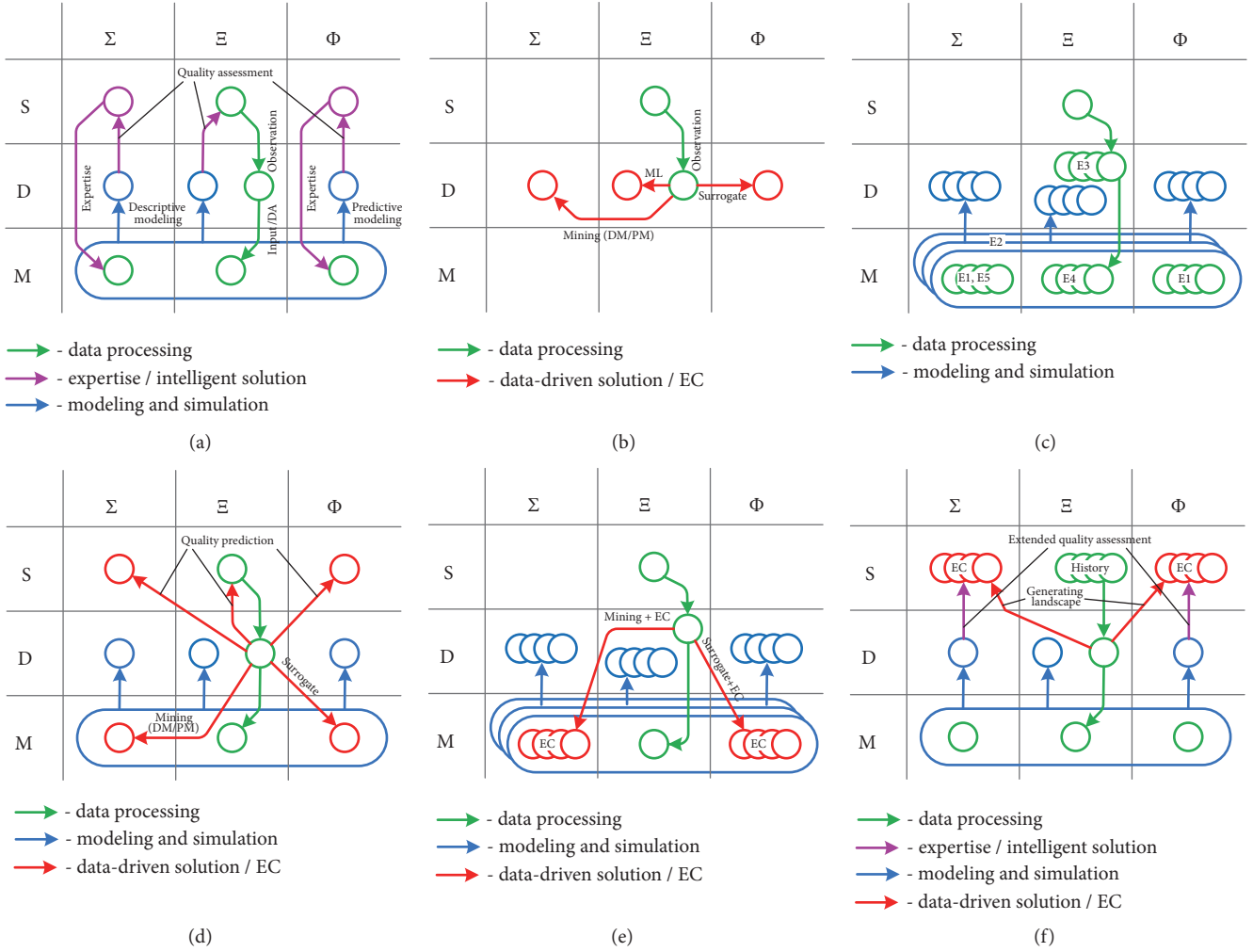


FIGURE 2: Complex modeling patterns: (a) regular modeling; (b) data-driven modeling; (c) ensemble-based modeling; (d) data-driven support of complex modeling; (e) EC in hybrid complex modeling; (f) evolutionary space discovery in hybrid complex modeling.

ensemble, alternative models ensemble, data-driven ensemble, parameter diversity ensemble, and metaensemble. All these patterns can be applied within a context of the proposed framework. Still, an extension of ensemble structure increases structural complexity of the model and thus leads to the need for additional (automatic) control procedures. Moreover, the performance issues of P1 are getting even worthier in ensemble modeling.

P4. One of the key ideas of the proposed approach is an implementation of data-driven analysis of model states, structure, and behavior. To implement it within a conceptual framework we propose pattern for data-driven complex modeling (Figure 2(d)). It includes identification and prediction of a model structure through DM and PM techniques ($\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Sigma}$) and generation of surrogate models for injection into the complex model ($\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Phi}$). In addition, it is possible to use data-driven techniques to predict the quality of the considered model and use it for model optimization ($\Gamma_{D \rightarrow S}^{\Xi}$, $\Gamma_{D \rightarrow S}^{\Xi \rightarrow \Sigma}$, and $\Gamma_{D \rightarrow S}^{\Xi \rightarrow \Phi}$).

P5. A key pattern for EC implementation is presented in Figure 2(e). Here EC is used to identify a model structure ($\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Sigma}$) and surrogate submodels ($\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Phi}$) with a consideration of population of models. As a result, modeling result is also (as well as in P3) presented in multiple instances which may be analyzed, **filtered and evolved** within consequent iterations over changing time (and processing of coming observations of the system) or within a single timestamp (and fixed observation data).

P6. Finally, last presented pattern (Figure 2(f)) is aimed at investigation of system phase space using DD-models and/or EC to reflect unobservable landscape for estimation of model positioning, assessing its quality in inferring of (sub-)optimal structural ($\Gamma_{D \rightarrow S}^{\Xi \rightarrow \Sigma}$ and $\Gamma_{M \rightarrow D}^{\Sigma}$) and functional ($\Gamma_{D \rightarrow S}^{\Xi \rightarrow \Phi}$ and $\Gamma_{M \rightarrow D}^{\Phi}$) characteristics of the actual system.

These patterns could be easily combined to obtain better results within a specific application. Especial interest from the point of view of EC is attracted to the patterns where a set of models (or sub-model) instances is considered (P5,

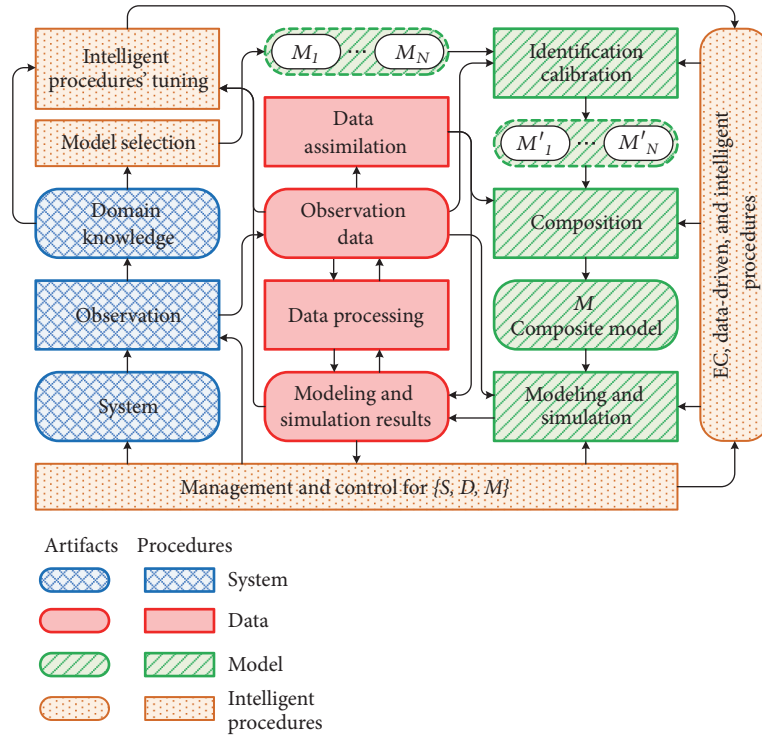


FIGURE 3: Artifacts and procedures within a typical composite solution.

P6). It is possible to consider ensemble-based techniques (P3) in a fashion of EC, but within our approach, we prefer consideration of ensemble as a composite model with several submodels. In that case, ensemble management refers to the concept of complex model structure.

Several important goals may be reached within the presented patterns:

- (i) automation of complex model management with intelligent solutions, DD-models, and EC;
- (ii) optimization of model structure and application under defined limitations in precision and performance;
- (iii) enhanced ways of domain knowledge discovery for applications and general investigation of a system.

2.3. Composite Solution Development. The proposed structure of core concepts and patterns may be applied in various ways to form a solution which combine operators with original implementation within the solutions or implemented as external model calls. Figure 3 shows the essential elements (artifacts and procedures) in a typical composite solution within the proposed conceptual layers (S , D , and M). S -layer includes actual system's state which can be assessed through the observation procedure and described by explicit domain knowledge. D -layer includes datasets divided into observation data and simulation/modeling data with procedures for data processing and data assimilation. Finally, M -layer includes a set of available basic models $M_1 \dots M_N$ which may be identified, calibrated with available data having tuned

models $M'_1 \dots M'_N$ as a result. Here, essential elements are model composition (which may be performed either automatically, or by the modeler) and application of the model.

The key benefit of the approach is an application of a combination of EC, data-driven and intelligent procedures to manage the whole composite solution including data processing, modeling, and simulation to lower uncertainty in $\Sigma \times \Xi \times \Phi$. Within the shown structure these procedures may be applied:

- (i) to rank and select alternative models;
- (ii) to support model identification, calibration, composition, and application;
- (iii) to manage artifacts on various conceptual layers in a systematic way;
- (iv) to infer implicit knowledge from available data and explicitly presented domain knowledge.

The shown example draws a brief view on the composite solution development while the particular details may differ depending on a particular application. Key important procedures within the proposed composite solution are the implementation of intelligent procedures to support model identification and systematic management of composite model are considered in Sections 2.4 and 2.5.

2.4. Evolutionary Model Identification. Implementing evolution of models within a complex modeling task structure, functional and quantitative parameters are usually considered as genotype whereas model output (data layer) are

considered as phenotype. Within the proposed approach we can adapt basic EC operations definition within genotype-phenotype mapping [13]:

- (i) epigenesis as model application: $f_1 : S \times M \rightarrow D$;
- (ii) selection: $f_2 : S \times D \rightarrow D$;
- (iii) genotype survival: $f_3 : S \times D \rightarrow M$;
- (iv) mutation: $f_4 : M \rightarrow M$.

In addition, we consider quality assessment usually treated as fitness for selection and survival (or in more complex algorithms for controlling of other operations like mutation):

- (i) data quality: $q_d : S \times D \rightarrow Q_d$;
- (ii) model quality: $q_m : S \times M \rightarrow Q_m$.

Here Q_d and Q_m are often considered as \mathbb{R}^N with some quantitative quality metrics. Model quality usually are considered through data quality, i.e., $q_m \sim q_d(s, f_1(s, m))$, but within our approach this separation is considered as important because in addition we introduce supporting operations with data-driven procedures as in complex modeling many of these functions (first of all f_1 , q_m , and q_d) have significant difficulties to be applied directly (some of these issues are considered in relationship with patterns). Data driven operations (first of all, f_1 and q_m) can be introduced as substitution of previously introduced basic operations (see also patterns P2, P4, and P6):

- (i) epigenesis as DD-model application: $f_1^d : S \times M \rightarrow D$;
- (ii) model generation: $g^d : S \times D \rightarrow M$;
- (iii) model quality prediction: $q_m^d : S \times M \rightarrow Q_m$;
- (iv) space discovery: $w^d : S \times D \rightarrow S$.

Operation w^d could be used within an intelligent extension within selection or survival operations (f_2 and f_3). It becomes especially important in case of lack of knowledge in system's structure or functional characteristics. Operation g^d at the same time could be used as a part of mutation operation f_4 (or initial population generation). Having this extension, we can implement enhanced versions of EC algorithms (e.g., genetic algorithms, evolution strategies, and evolutionary programming) with data-driven operations to overcome or at least to lower complex modeling issues.

2.5. Model Management Approach and Algorithm. By model management we assume operations with models within problem domain solution development and application. This includes identification, calibration, DA, optimization, prediction, and forecasting. To systematize the model management in the presented patterns we propose an approach for explicit consideration of spaces S , D , and M within hybrid modeling with EC and DD-modeling. To summarize complex modeling procedures within the approach, we developed a high-level algorithm which includes series of steps to be implemented within a context of complex model management.

Step 1 (space discovery). This step identifies the description of phase space (in most cases, S) in case of lack of knowledge or for automation purposes. For example, the step could be applied in the discovery of system state space or model structure. Space description may include (a) distance metrics; (b) proximity structure (e.g., graph, clustering hierarchy, and density); (c) positioning function. One of the possible ways to perform this step is an application of DM and EC algorithm to available data (see pattern P6).

Step 2 (identification of supplementary functions). Data-driven functions (Φ) are applied to work in model evolution with consideration of space (landscape) representation as available information.

Step 3 (evolutionary processing of a set of models). This step is described by a combination of basic EC operations (population initialization, epigenesis, selection, mutation, and survival) with supplementary functions. A form of combination depends on (a) selected EC algorithm; (b) application requirements and restrictions; (c) model-based issues (e.g., performance, quality of surrogate models, etc.).

Step 4 (assimilation of updated data and knowledge). This step is applied for automatic adaptation purposes and implement DA algorithm. DA can be applied to (a) set of models, (b) EC operations (e.g., affecting selection function); (c) supplementary functions (as they are mainly data-driven); (d) phase space description (if descriptive structure is identified from changed data or/and knowledge).

The steps can be repeated in various combination depending on an application and implemented pattern. Also, the steps are general and could be implemented in various ways. Several examples are provided in the Section 3.

2.6. Available Building Blocks of a Composite Solutions. EC proposes a flexible and robust solution to identify complex model structures within a complex landscape with possible adaptation towards changing condition and system's state (including new states without prior observation). A significant additional benefit is an ability to manage alternative solutions simultaneously with possible switching and various combination of them depending on the current needs. Still, within the task of model identification and management, the EC (and also many metaheuristics) have certain drawbacks which require additional steps to implement the approach within particular conditions:

- (i) high computational cost due to the multiple runs of a model;
- (ii) low reproducibility and interpretability of obtained results due to randomized nature of the searching procedure;
- (iii) complicated tuning of hyperparameters for better EC convergence;
- (iv) indistinct definition of genotype boundaries;

- (v) complicated mapping of genotype to phenotype space.

To overcome these issues, the proposed approach involves two options. First, the intelligent procedures may be used to tune EC hyperparameters (P5), predict features of genotype-phenotype mapping, boundaries, etc. (P4), and discover interpretable states and filters (for system, data, and model) to control convergence and adaptation of population (P2, P4, and P5) with interpretable and reproducible (through the defined control procedure). Second, the composite model may use various approaches, methods, and elements to obtain better quality and performance of the solution:

- (i) surrogate models (P2, P4, P5) which may increase performance (for example, within preliminary and intermediate optimization steps);
- (ii) ensemble models (P3) which may be considered as interpretable and controllable population;
- (iii) interpretation and formal inference using explicit domain-specific knowledge and results of data mining to feed procedures of EC and infer parameters in both models and EC.
- (iv) controllable space decomposition (P6) with predictive models for possible areas and directions of population migration in EC to explicitly lower uncertainty and obtain additional interpretability;

Finally, an essential feature of the proposed approach is a holistic analysis of a composite solution with possible coevolution models (submodes within a composite model) and data processing procedures.

3. Application Examples

This section presents several practical examples where the proposed approach, patterns, or some of their elements were applied. The examples were intentionally selected from diverse problem domains to consider generality of the approach. The considered problems are developed in separated projects which are in various stages. Problem #1 (ensemble metocean simulation) was investigated in a series of projects (see, e.g., [11, 14, 15]). Within this research we are trying to extend model calibration and DA with EC techniques to develop more flexible and accurate multimodel ensembles. Problem #2 (clinical pathways (CPs) modelling) is important in several ongoing project aimed to model-based decision support in healthcare (see, e.g., [16–18]). The proposed approach plays important role by enabling deeper analysis of clinical pathways in various scenarios (interactive analysis of available CPs with identification of clusters of similar patients, DA in predictive modelling of ongoing cases, etc.). Finally, Problem #3 shows very early results in recently started project in online social network analysis.

3.1. Problem #1: Evolution in Models for Metocean Simulation. The environmental simulation systems usually contain

several numerical models serving for different purposes (complementary simulation processes, improving the reliability of a system by performing alternative results, etc.). Each model typically can be described by a large number of quantitative parameters and functional characteristics that should be adjusted by an expert or using intelligent automatized methods (e.g., EC). Alternative models inside the environmental simulation system can be joined in ensemble according to complex modeling pattern based on evolutionary computing (a combination of P3 and P5 patterns). In the current case study, we introduce an example illustrated an ensemble concept in forms of the alternative models ensemble, parameter diversity ensemble, and metaensemble. For identification of parameters of proposed ensembles (in a case of model linearity) least square method or (in a case of nonlinearity) optimization methods can be used. As we need to take into account not only functional space Φ and space of parameters Ξ for a single model but also perform optimal coexistence of models in the system (i.e., Σ), evolutionary and coevolutionary approaches seem to be an applicable technique for this task. It is worth mentioning that coevolutionary approach can be applied to independent model realizations through an ensemble as a connection element. In this case parameters (weights) in the ensemble can be estimated separately from the coevolution procedure in a constant form or dynamically. As a case study of complex environmental modeling we design ensemble model that consists of the SWAN (<http://swanmodel.sourceforge.net/>) model for ocean wave simulation based on two different surface forcings by NCEP (<https://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.html>) and ERA Interim (<https://www.ecmwf.int/en/forecasts/datasets/archive-datasets/reanalysis-datasets/era-interim>). Thus, different implementations of SWAN model were connected in the form of an alternative models ensemble with least-squares-calculated coefficients defining structure of the complex model. Two parameters—wind drag and whitecapping rate (WCR)—were calibrated using evolutionary and coevolutionary algorithms implementing $\Gamma_{D \rightarrow M}^{\Xi \rightarrow \Phi}$ in P5 (for detailed sensitive analysis of SWAN see [19]). Case of coevolutionary approach can be represented in a form of parameter diversity ensemble, where each population is constructed an ensemble of alternative model results with different parameters. Also, we can add ensemble weights to model parameters diversity and get metaensemble that can be identified in a frame of coevolutionary approach.

In a process of model identification and verification, measurements from several wave stations in Kara sea were used. Fitness function represents the mean error (RMSE) for all wave stations. For results verification MAE (mean absolute error) and DTW (dynamic time wrapping) metrics were used.

Figure 4(a) represents surface (landscape) of RMSE in the space of announced parameters (drag and WCR) for implementations SWAN+ERA and SWAN+NCEP. It can be seen that the evolutionary-obtained results converge to the minimum of possible error landscape. The landscape was obtained by starting the model with all parameters variants

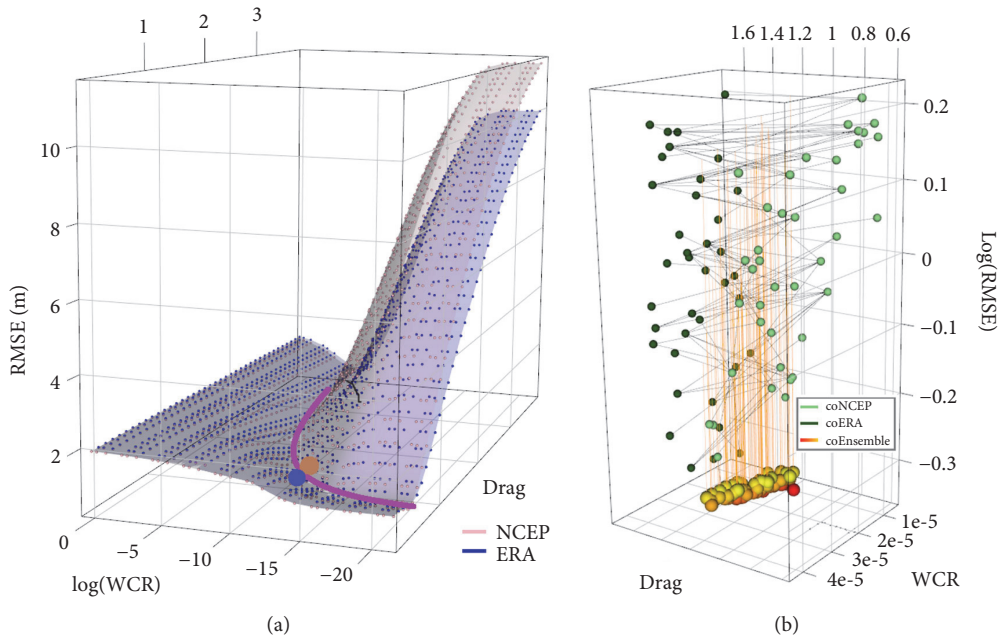


FIGURE 4: Metocean simulation: (a) error landscape for wave height simulation results using ERA and NCEP reanalysis as input data and (b) Pareto frontier of coevolution results for all generations.

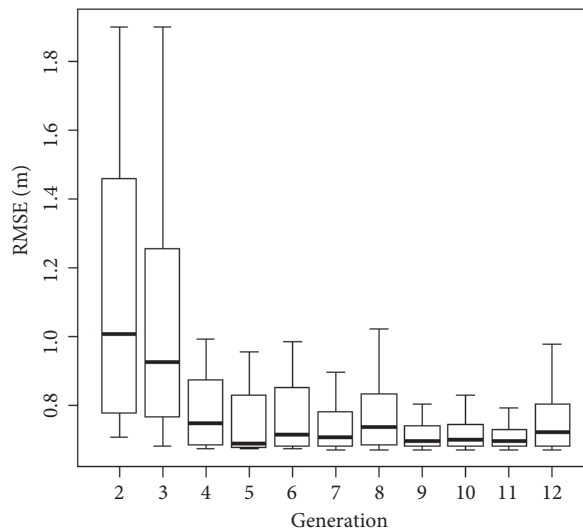


FIGURE 5: Coevolution convergence of diversity parameters ensemble for metocean models.

from full 30x30 grid (i.e., 900 runs), while evolutionary algorithm was converged in 5 generations with 10 individuals (parameters set) in population (50 runs) that allows performing identification two orders faster. The convergence of co-evolution for SWAN+ERA+NCEP case is presented in Figure 5.

Although error landscapes for a pair of implementations SWAN+ERA and SWAN+NCEP are close to each other, separated evolution does not consider optimization of ensemble result. For this purpose, we apply coevolutionary approach that produces the set of Pareto-optimal solutions for each generation. Figure 4(b) shows that the error of each model in

the ensemble is significant (coNCEP and coERA for models along), but the error of the whole ensemble (coEnsemble) converges to minimum very fast.

Obtained result can be analyzed from the uncertainty reduction point of view. Model parameters optimization helps to reduce parameters uncertainty that can be estimated through error function. But when we apply an ensemble approach to evolutionary optimized results, it is suitable to talk about reduction of the uncertainty connected with input data sources (NCEP and ERA) as well. Moreover, metaensemble approach allowed reduction of uncertainty, connected with ensemble parameters.

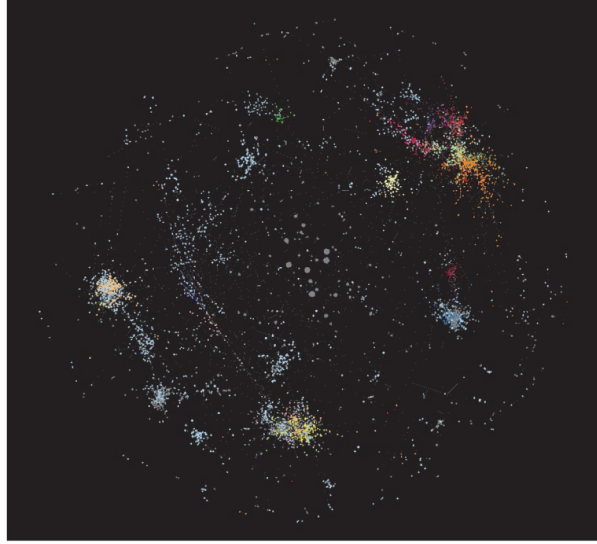


FIGURE 6: Graph-based representation of processes space in healthcare (interactive view) (Demonstration available at <https://www.youtube.com/watch?v=EH74f1w6EeY>).

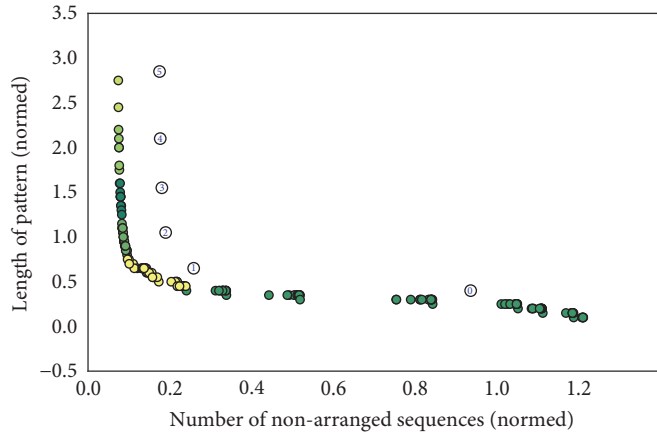
Summarizing results of the metocean case study we can denote that EC approach shows significant efficiency up to 120 times compared with grid search without accuracy losses. According to this experimental study, quality of ensemble with evolutionary optimized models is similar to results of the grid search and MAE metric is equal to 0.24 m and DTW metric – 51. Also, we can mention that coevolutionary approach provides 10 % accuracy gain compared with results of single evolution of model implementations, but this is still similar to ensemble result with evolutionary optimized models. Nevertheless, coevolutionary approach allowed to achieve 200 times acceleration. Within the context of the proposed approach space Φ were investigated using defined structure of the model in space Σ for the purpose of model calibration.

3.2. Problem #2: Modeling Health Care Process. Modeling healthcare processes are usually related to the enormous uncertainty and variability even when modeling single disease. One of the ways to identify a model of such process is PM [20]. Still, direct implementation of PM methods does not remove a major part of the uncertainty. Within current research, we applied the proposed approach for identification purposes both in the analysis of historical cases and prediction of single process development. Here we consider processes of providing health care in acute coronary syndrome (ACS) cases which is usually considered as one of the major death causes in the world. We used a set of 3434 ACS cases collected during 2010-2015 in Almazov National Medical Research Centre one of the leading cardiological centers in Russia. The data set contains electronic health records of these patients with all registered events and characteristics of a patient.

To simplify consideration of multidimensional space of possible processes ($\Gamma_{D \rightarrow S}^{\Sigma} \Gamma_{S \rightarrow D}^{\Sigma}$ for analysis of Σ on layer S)

we introduced graph-based representation of this space with vertices representing cases and edges representing proximity of cases. Analysis of such structure enables easy discovering of common cases (e.g., as communities in graph). Such discovering enables explicit interpretable structuring of the space and representation of further landscape for EC in terms of P6 pattern. Moreover, direct interactive investigation of visual representation of such structure (see Figure 6) provides significant insights for medical researchers.

We have developed evolutionary-based algorithm for patterns identification and clustering in such representation with two criteria to be optimized (see Figure 7). Here processes were represented by a sequence of labels (symbols) denoting key events in PM model. Typical patterns were then selected for Pareto frontier. The convergence process is demonstrated in Figure 8 (10 best individuals from Pareto frontier according to the integral criterion were selected). As a result, this solution may refer to P5 pattern and operator $\Gamma_{D \rightarrow M}^{\Sigma}$ while discovering model structure. Figure 9 shows an example of typical process model (i.e., structural characteristic of the model) for one of the identified clusters. Detailed description of the approach, algorithms, and results on CPs discovering, clustering, and analysis including comparison of three version of CP discovery algorithms with performance comparison can be found in [10]. An important outcome of the approach being applied in this application is interpretability of the clusters and identified patterns. For example, 10 clusters and corresponding CPs obtained interpretation by cardiologists from Almazov National Medical Research Centre. The obtained interpretation and further discovering and application with CP structure are presented in [17]. Another important benefit given by such space structure discovering is lowering uncertainty of patient's treatment trajectory by a hierarchical positioning of an evolved process (selection of a cluster and selection of position within the cluster). For example, discrete-event simulation model



- 0 AFIFNE
- 1 AFNIFEIFDNE
- 2 AFENIFEENIFEDNFIEFE
- 3 AFNIFEDINIFENDDFEIDNFEEDFIFE
- 4 AEFNINFEDEFNEIDFDFNEFDIEDNFNEIFDDNIEFFDE
- 5 AFANIFADIFEFNIEFDEDEINFEFDEDFFDNIFENDFIEIDAFEFEDEIFEDNA

FIGURE 7: Pareto frontier for CP patterns discovery.

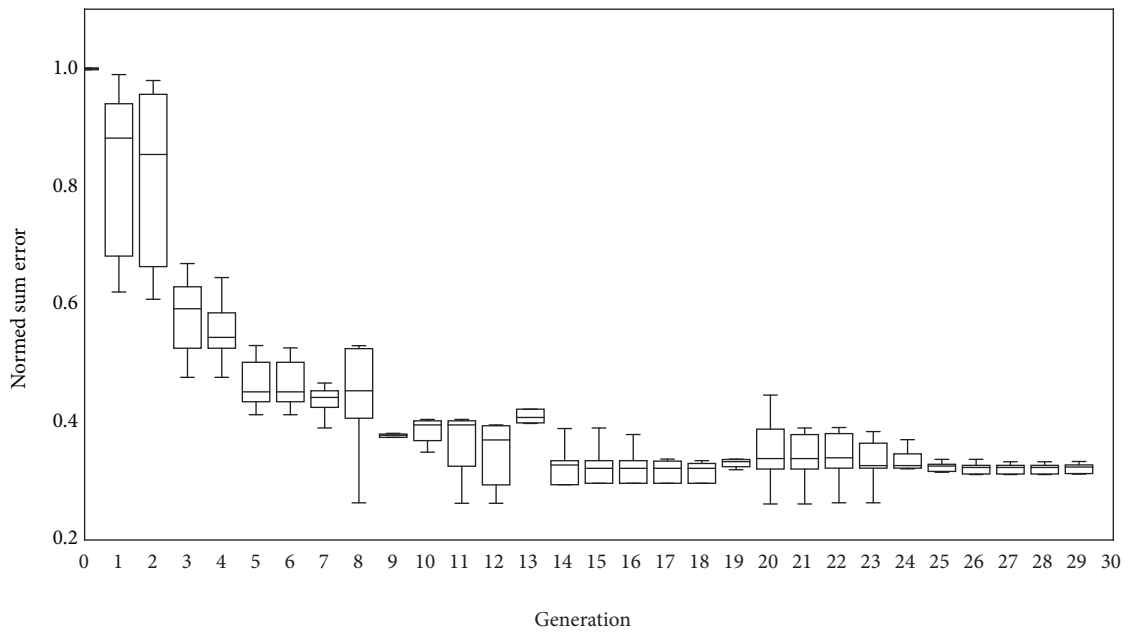


FIGURE 8: Evolutionary convergence during CP pattern discovery.

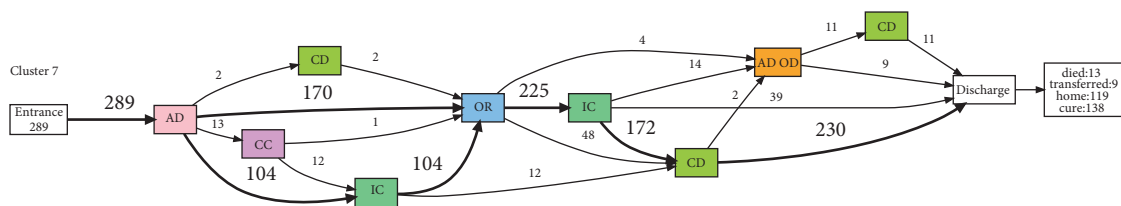
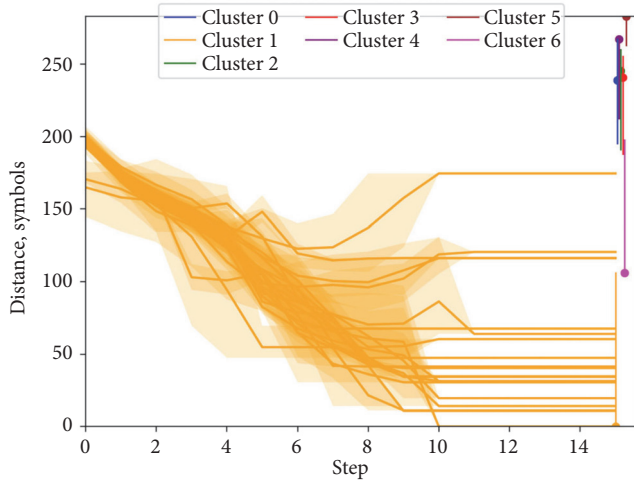
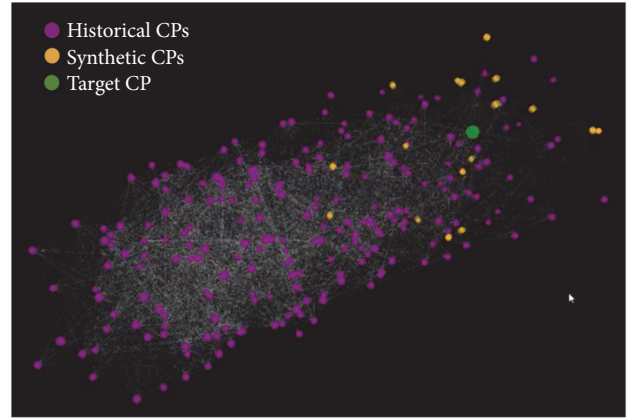


FIGURE 9: Example of process model showing transfers between hospital's departments.

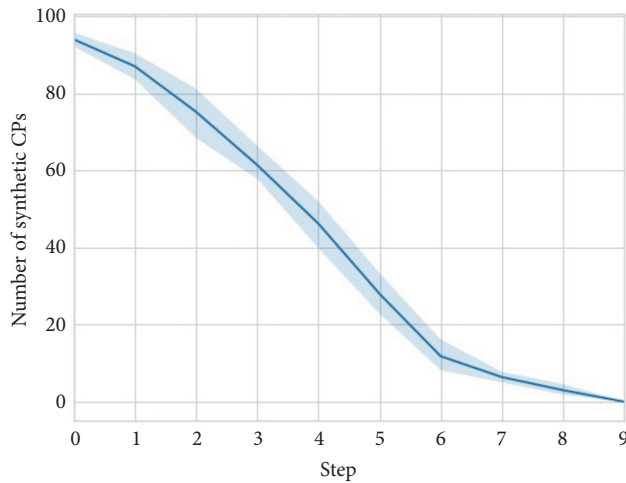
died:13
transferred:9
home:119
cure:138



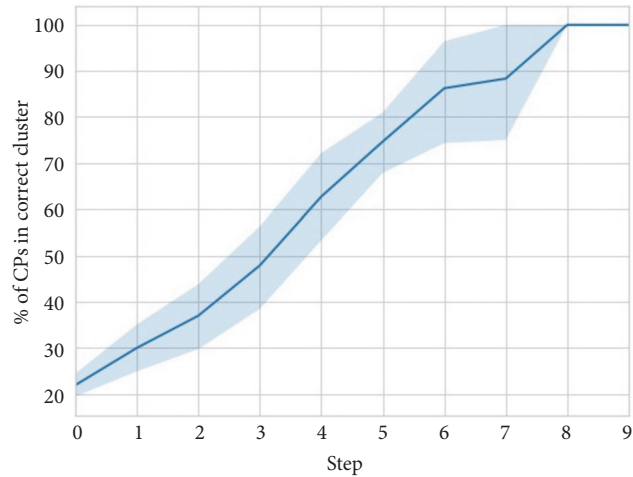
(a)



(b)



(c)



(d)

FIGURE 10: Evolution of synthetic CPs: (a) CP population convergence; (b) evolution of possible CP (demonstration available at <https://www.youtube.com/watch?v=twvfX9zKsY8>); (c) number of synthetic CPs; (d) % of CPs in correct cluster.

described in [17] provides a more appropriate length of stay distribution within simulation with discovered classes of CPs (Kolmogorov-Smirnov statistics decreased by 51% (from 0.255 to 0.124)).

Furtherly we propose an algorithm to dynamically generate possible development of the process in healthcare using identified graph-based space representation with evolutionary strategies, assimilating incoming data (events) within a case ($\Gamma_{M \rightarrow D}^{\Sigma}$ in P5 and $\Gamma_D^{\Sigma \rightarrow \Sigma}$ in P2). We consider convergence (Figure 10(a)) of the introduced synthetic continuation of the processes to the right class (identified clusters of typical cases were used) with mapping to the graph-based space representation with proximity measures (Figure 10(b)). As a result, the appearance of the CP's events decreases the number of synthetic CPs and increases percentage of CPs positioned in the correct cluster (see an example in Figure 10(c) and Figure 10(d) correspondingly). This enables interpretable positioning and uncertainty lowering in predicting further CP's development for a particular patient.

Here a combination of patterns P2, P5, and P6 in the implementation of the proposed algorithm (see Section 2.5) enables interactive investigation of processes space and data assimilation into a population of possible continuations of a single process during its evolving. This solution can be applied in exploratory modeling and simulation of patient flow processing as well as decision support in specialized medical centers.

3.3. Problem #3: Mining Social Media. Nowadays social media analysis (that began with static network models emphasizing a topology of connections between users) strives to explore dynamic behavioral patterns of individuals which can be recovered from their digital traces on the web. The prediction of social media activities requires combining analytical and data-driven models as well as identifying the optimal structure and parameters of these models according to the available data. Here we show an example of the problem in this field involving evolutionary identification of a model.

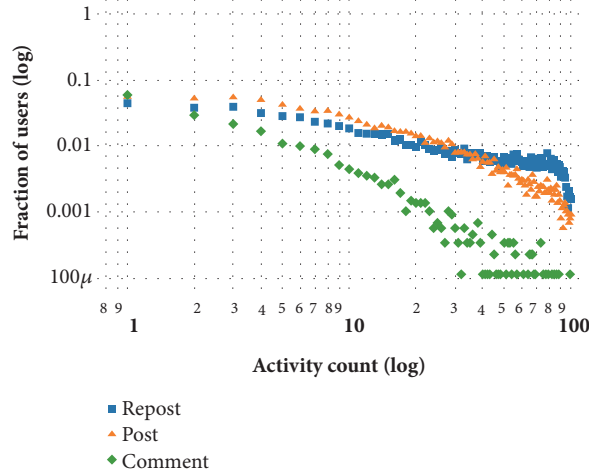


FIGURE 11: Distribution of posts, reposts, and comments on personal walls of subscribers of bank community.

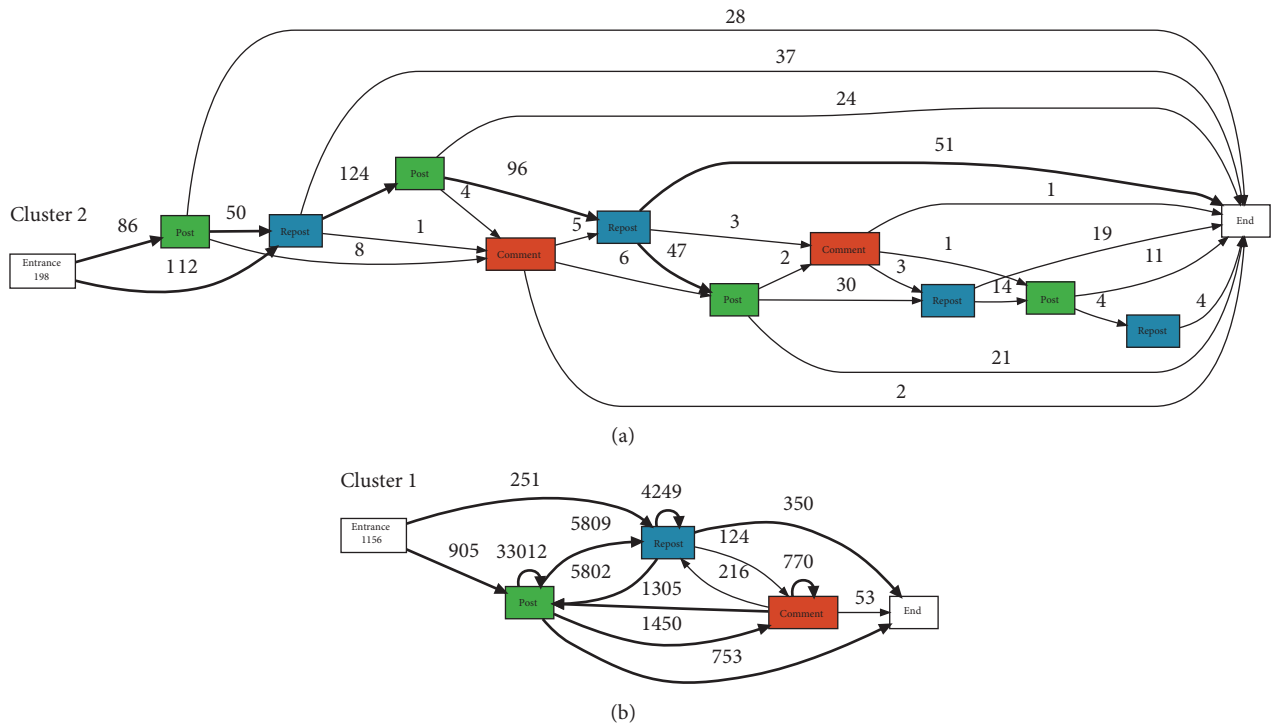


FIGURE 12: Example of process model (a) with expanded cycles and (b) with collapsed cycles.

A digital trace of a user in an online social network (OSN) is a sequence (chain) of observed activities separated with time gaps. Each OSN supports different types of “hidden” and observable activities. For example, in a largest Russian social network vk.com (further is denoted as VK) a user has a personal page (wall) with three types of activities: post (P)—when a user makes a record by himself; repost (R)—when the user copies the record of another user or community to his or her wall and comment (C)—when the user comments the record on his or her wall. Figure 11 illustrates the distribution of these activities for subscribers of large Russian bank community in VK. The collected dataset

consists of 100 (or less if unavailable) last entries (posts or reposts), and comments for the entries for 8K user walls in a period January 2017–December 2017. Comments are much less common than posts and reposts. The distributions of the posts and reposts are similar, but there is a group of “spreaders” with a significant number of reposts.

We applied the technique described in Section 3.2 to analyze the processes. Still, the considered process has significantly different structure. By default, it is continuous with random repetition of events, while healthcare process in ACS cases has finite and more “strong” structure. Figure 12 shows a typical process structure identified with EC-based approach

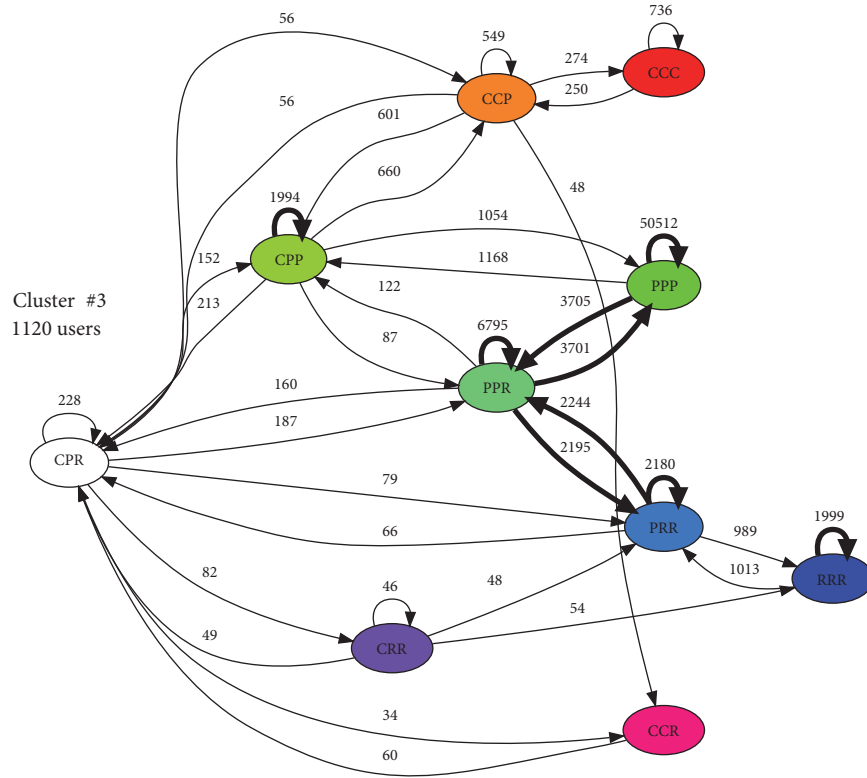


FIGURE 13: Example of process model for cluster #3 using n-gram analysis.

TABLE 1: Mean of activities' combinations for users' clusters.

Cluster	Size	CCC	CCP	CCR	CPP	CPR	CRR	PPP	PPR	PRR	RRR
1	5238	0.52	0.74	0.24	1	0.72	0.42	6.68	6.4	7.56	8.53
2	2110	0.11	0.13	0.16	0.16	0.36	0.59	2.09	3.95	12.32	60.3
3	1120	0.91	1.38	0.14	3.62	0.75	0.18	50.02	11.78	5	2.77

and visualized with expanded cycles (a) and with collapsed cycles (b). The second one could be considered as more relevant than the first one which is significantly affected by a length of selected history. It is natural to consider it as a random process or state-transition model. In that case, three identified clusters (characterized by various frequencies of transitions) could be interpreted as typical behavior models.

N-grams analysis is often used to detect patterns in people's behaviors [21, 22]. N-grams analysis is based on counting frequencies of combinations or sequences of activities. We collected all sorted 3-grams (so called 3-sets) for each user's sequence to analyze the frequency of event combinations. As a result, three clusters of vectors with 3-sets chains were identified with k-means clustering method. Figure 13 shows all combinations and transitions between them for cluster #3 as an example. Using Figure 12 and Table 1, it is possible to see that cluster #3 includes users who often make new records (P) and sometimes comment records (C). So, cluster #3 mostly consists of "bloggers." Cluster #2 includes "spreaders" who copy other records (R) frequently. And the biggest cluster #1 consists of people who make new records and copy other ones equally but less intensively comparing to other clusters.

That may be considered as a typical behavior for user of OSN. N-grams analysis allows detecting typical behavioral patterns and obtaining process models for social media activities using chains of different lengths as input data. Thus, this type of data-driven modeling is more appropriate to research continuous processes. Figure 14 shows a graph-based representation of process space with of all users' patterns.

This subsection provides very early results. Next step within application of the proposed approach in this application includes an extension of process model structure (a) with temporal labeling (gaps between events); (b) considering process within a sliding time window to get more structured processes; (c) linking the model with causal inference; (d) introduction of DM techniques for EC positioning of ongoing processes in model space. We believe that these extensions could enhance discovery of model structure (P4) and provide deeper insight on social media activity investigation.

4. Conclusion and Future Work

The development of the proposed approach is still an ongoing project. We aimed for further systematization and detailing

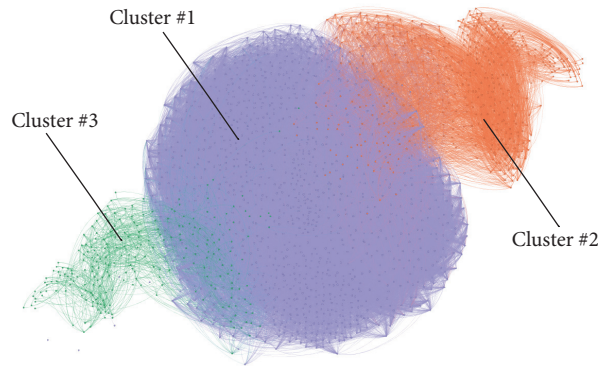


FIGURE 14: Graph-based representation of processes' space for three clusters in social media activity.

of the proposed concepts, methods, and algorithms, as well as more comprehensive and deeper implementation of EC-based applications. Further work of the development includes the following directions:

- (i) dualization on the role of data-driven and intelligent operations in proposed approach and described patterns;
- (ii) extended analysis of various EC techniques applicable within the approach;
- (iii) investigation on EC-based discovery for models of complex systems with lack or inconsistent observations;
- (iv) detailed formalization of expertise and knowledge-based methods within the approach;
- (v) extending the approach with interactive user-centered modelling and phase space analysis;
- (vi) development of multilayered approach for decision support and control of system and process S , available data D , and complex model M .

Data Availability

The data used in the examples presented in Section 3 were initially obtained within complimentary projects performed by the authors of the paper. The data is available from the corresponding author upon request after explicit claiming of the purpose and plan of requested data usage to check for possible violation of the corresponding projects' rules.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper presents an extension and further development of the work [23]. This research is financially supported by The Russian Scientific Foundation, Agreement #14-11-00823 (15.07.2014).

References

- [1] N. Boccarda, *Modeling complex systems*, Graduate Texts in Physics, Springer, New York, Second edition, 2010.
- [2] H. McManus and D. Hastings, "A framework for understanding uncertainty and its mitigation and exploitation in complex systems," *IEEE Engineering Management Review*, vol. 34, no. 3, pp. 81–94, 2006.
- [3] W. Walker, P. Harremoës, J. Rotmans et al., "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support," *Integrated Assessment*, vol. 4, no. 1, pp. 5–17, 2003.
- [4] J. Yan and J. R. Deller, "NARMAX model identification using a set-theoretic evolutionary approach," *Signal Processing*, vol. 123, pp. 30–41, 2016.
- [5] I. G. Kevrekidis, C. W. Gear, and G. Hummer, "Equation-free: The computer-aided analysis of complex multiscale systems," *AIChE Journal*, vol. 50, no. 7, pp. 1346–1355, 2004.
- [6] H. Ishaish, A. Cortés, and M. A. Senar, "Parallel Multi-level Genetic Ensemble for Numerical Weather Prediction Enhancement," *Procedia Computer Science*, vol. 9, pp. 276–285, 2012.
- [7] G. Dumedah, "Formulation of the Evolutionary-Based Data Assimilation, and its Implementation in Hydrological Forecasting," *Water Resources Management*, vol. 26, no. 13, pp. 3853–3870, 2012.
- [8] V. V. Kashirin, A. A. Lantseva, S. V. Ivanov, S. V. Kovalchuk, and A. . Boukhanovsky, "Evolutionary simulation of complex networks' structures with specific functional properties," *Journal of Applied Logic*, vol. 24, no. part A, pp. 39–49, 2017.
- [9] S. V. Kovalchuk, P. A. Smirnov, K. V. Knyazkov, A. S. Zagarskikh, and A. V. Boukhanovsky, "Knowledge-Based Expressive Technologies Within Cloud Computing Environments," in *Practical Applications of Intelligent Systems*, vol. 279 of *Advances in Intelligent Systems and Computing*, pp. 1–11, Springer, Berlin, Germany, 2014.
- [10] A. A. Funkner, A. N. Yakovlev, and S. V. Kovalchuk, "Towards evolutionary discovery of typical clinical pathways in electronic health records," *Procedia Computer Science*, vol. 119, pp. 234–244, 2017.
- [11] S. V. Kovalchuk and A. V. Boukhanovsky, "Towards Ensemble Simulation of Complex Systems," *Procedia Computer Science*, vol. 51, pp. 532–541, 2015.

- [12] S. V. Kovalchuk, A. V. Krikunov, K. V. Knyazkov, and A. V. Boukhanovsky, "Classification issues within ensemble-based simulation: application to surge floods forecasting," *Stochastic Environmental Research and Risk Assessment*, vol. 31, no. 5, pp. 1183–1197, 2017.
- [13] D. Fogel, "Phenotypes, genotypes, and operators in evolutionary computation," in *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, p. 193, Perth, WA, Australia.
- [14] S. V. Ivanov, S. V. Kovalchuk, and A. V. Boukhanovsky, "Workflow-based Collaborative Decision Support for Flood Management Systems," *Procedia Computer Science*, vol. 18, pp. 2213–2222, 2013.
- [15] A. Gusarov, A. Kalyuzhnaya, and A. Boukhanovsky, "Spatially adaptive ensemble optimal interpolation of in-situ observations into numerical vector field models," in *Proceedings of the 6th International Young Scientist Conference on Computational Science, YSC 2017*, pp. 325–333, Finland, November 2017.
- [16] A. V. Krikunov, E. V. Bolgova, E. Krotov, T. M. Abuhay, A. N. Yakovlev, and S. V. Kovalchuk, "Complex data-driven predictive modeling in personalized clinical decision support for Acute Coronary Syndrome episodes," in *Proceedings of the International Conference on Computational Science, ICCS 2016*, pp. 518–529, USA, June 2016.
- [17] S. V. Kovalchuk, A. A. Funkner, O. G. Metsker, and A. N. Yakovlev, "Simulation of patient flow in multiple healthcare units using process and data mining techniques for model identification," *Journal of Biomedical Informatics*, vol. 82, pp. 128–142, 2018.
- [18] A. Yakovlev, O. Metsker, S. Kovalchuk, and E. Bologova, "Prediction of in-hospital mortality and length of stay in acute coronary syndrome patients using machine-learning methods," *Journal of the American College of Cardiology*, vol. 71, no. 11, p. A242, 2018.
- [19] A. Nikishova, A. Kalyuzhnaya, A. Boukhanovsky, and A. Hoekstra, "Uncertainty quantification and sensitivity analysis applied to the wind wave model SWAN," *Environmental Modelling & Software*, vol. 95, pp. 344–357, 2017.
- [20] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, "Process mining in healthcare: A literature review," *Journal of Biomedical Informatics*, vol. 61, pp. 224–236, 2016.
- [21] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg, "Your click decides your fate: Inferring Information Processing and Attribution Behavior from MOOC Video Clickstream Interactions," in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pp. 3–14, Doha, Qatar, October 2014.
- [22] C. Marceau, "Characterizing the Behavior of a Program Using Multiple-Length N-Grams," Defense Technical Information Center, 2005.
- [23] S. V. Kovalchuk, O. G. Metsker, A. A. Funkner et al., "Towards management of complex modeling through a hybrid evolutionary identification," in *Proceedings of the the Genetic and Evolutionary Computation Conference Companion*, pp. 255–256, Kyoto, Japan, July 2018.

Research Article

Data-Driven Finite Element Models of Passive Filamentary Networks

Brian Adam and Sorin Mitran 

Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3250, USA

Correspondence should be addressed to Sorin Mitran; mitran@unc.edu

Received 6 July 2018; Accepted 20 September 2018; Published 1 November 2018

Academic Editor: Francisco Chinesta

Copyright © 2018 Brian Adam and Sorin Mitran. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A data-driven procedure is introduced to construct finite element models of heterogeneous systems for which an accurate microscopic description is available. A filter to define coarsened finite element nodal values is defined from the principal modes obtained by singular value decomposition of the microscopic data. The resulting finite element nodal values are subsequently used to reconstruct local linearization of the system behavior, defining drag and stiffness matrices for an overdamped system. The procedure is exemplified for an actin mesh described by Brownian dynamics and eight-node cuboid finite elements but is generally applicable with respect to both the microscopic model and the type of finite element approximation. In contrast to standard finite element formulations derived from hypotheses on assumed deformation behavior, the data-driven procedure introduced here is completely determined by the observed behavior be it obtained from simulations or experiment.

1. Introduction

Materials with heterogeneous microscopic structure arise in a wide variety of applications and are frequently encountered in nature. For example, the mechanical behavior of a biological cell is largely determined by a network of actin filaments that actively polymerize and depolymerize exerting force on the cellular membrane and leading to cellular motility [1]. Many cellular processes depend on remodeling the actin network configuration leading to diverse mechanical responses [2]. The filamentary actin (F-actin) network exhibits viscoelastic behavior that becomes nonlinear at large deformations [3]. Observed elastic moduli are strongly influenced by dynamically varying cross-linking between network filaments ranging from 0.03 Pa to 300 Pa depending on cross-linker concentration [4]. While the qualitative physics of such networks [4, 5] can identify regions within the phase space of concentrations of various components, detailed and quantitatively correct models of cellular mechanics will probably require numerical simulation. Various approaches to modeling of networks arising in biology are reviewed in [6, 7]. The main difficulty encountered in constructing

accurate models is that while the basic behavior of individual components within a cell are fairly well understood, the overall complexity renders such almost first-principle approaches exceedingly expensive. For example, Brownian dynamic methods have been attempted for entire cell simulation [8] or extraction of viscoelastic properties [9], at considerable computational effort. Such studies have identified network stiffening mechanisms [10], the crucial role played by cross-linkers [11] in establishing viscoelastic properties of the network, that compare favorably with the mechanics of networks reconstituted from observation [12].

Of particular interest here are finite element models for the actin mesh. One line of research [13] introduces various types of behavior for individual components of the network such as filament elements or cross-linkers. At the other end of the modeling resolution scale are finite element gel continuum [14] models based upon large-deformation mechanics and an assumed constitutive relationship. Related to these approaches, the procedure introduced in this work seeks to use detailed microscopic simulation to obtain a finite element model, since this is the resolution level at which the basic biophysical processes are well

defined, and obtain a homogenized model. In contrast to aforementioned approaches, neither is a constitutive relationship at the continuum scale assumed to be known, nor is the goal to construct a finite element model of each constituent element of the network. The approach consists in leveraging microscopic simulation (Brownian dynamics in this work) to extract an effective finite element model, defining the constitutive relationships underlying network response to forcing from available microscopic data. As described in more detail below, the approach combines a succession of data reduction procedures based upon projection in Euclidean spaces. Data from the microscopic model are filtered by the principal modes observed during the motion. The filtered data are used to define nodal deformation values. The principal modes are also reduced to smaller dimension to define effective drag and stiffness matrices describing the linearized viscoelastic behavior of the network. Final model coefficients are determined by fitting to observed behavior over an ensemble of network instances to account for variability in network configuration.

2. Methods

The Brownian dynamics (BD) model of [15] is adopted to describe the actin network. In this approach, the velocity $\mathbf{v}_i(t) = \dot{\mathbf{x}}_i(t)$ of actin monomer i is determined by the interaction potential $U_i(\mathbf{x})$, an overall repulsive potential $U_i^m(\mathbf{x})$, and a background stochastic force from solvent molecules ξ ,

$$\gamma \mathbf{v}_i = -\nabla_i U_i(\mathbf{x}) - \nabla_i U_i^m(\mathbf{x}) + \xi, \quad (1)$$

where $\mathbf{x} = (\mathbf{x}_1 \dots \mathbf{x}_N)^T \in \mathbb{R}^m$ is the vector of all $N = m/3$ monomer positions, $\nabla_i = \partial/\partial \mathbf{x}_i$ is the gradient with respect to position vector of monomer i , and the stochastic forcing is assumed to be Gaussian

$$\langle \xi(t) \xi(t') \rangle = 2\gamma k_B T \delta_{\alpha\beta} \delta(t - t'), \quad \langle \xi(t) \rangle = 0, \quad (2)$$

with indices α, β running over spatial components, γ the drag coefficient, k_B the Boltzmann constant, and T the temperature. The interaction potential U_i contains terms describing interactions of monomers within filaments (length and angle dependent) and a volume exclusion term of Lennard Jones form. The overall repulsive potential U_i^m is also of Lennard Jones form. No length and angle interaction terms act upon free actin monomers. For full details consult [15], including the model coefficients used in computations here.

The BD (1) form a nonlinear system of stochastic differential equations that accurately describes the processes of actin polymerization that leads to generation of mechanical force. For studies over a small part of a cell, such as in the immediate vicinity of a small portion of the cellular membrane in order to characterize the Brownian ratchet mechanism of cellular protrusion [16–20], direct numerical solution of the BD system (1) is feasible. However, for larger-scale studies of the actin mesh network over an entire cell, the computational cost becomes prohibitive, and a reduced model capturing essential features is required. In

particular, it is desirable to obtain a model of the constitutive relationship for the homogenized model informed by the accurate detailed BD model. Though the discussion here is presented for the particular case of the actin mesh network forming the cytoskeleton of a cell, the above situation is generic and also appears in consideration of various gels [5, 21] or the collagen extracellular matrix [22–24] among other applications.

From the nonlinear system describing time evolution of the mesh network

$$\dot{\mathbf{x}} = \mathbf{h}(t, \mathbf{x}, \xi), \quad (3)$$

a local linearization around the state \mathbf{x}_0 can be written as

$$\mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}. \quad (4)$$

We assume that $\mathbf{x}(t)$ describes deviation from the reference state \mathbf{x}_0 such that the time average over simulation time $T \langle \mathbf{x}(t) \rangle = T^{-1} \int_0^T \mathbf{x}(t) dt = \mathbf{x}_0$. Given the nonlinearity and possible history dependence of the network configuration, it can be expected that the drag and stiffness matrices depend on the chosen reference state, $\mathbf{D}(\mathbf{x}_0)$, $\mathbf{K}(\mathbf{x}_0)$, and the analysis presented below would need to be repeated for markedly different reference states. In the example presented here, an ensemble of reference states \mathbf{x}_0 is generated by simulating growth of a mesh from random placement of actin monomers within the computational domain until the average filament length ℓ reaches some fraction f of the simulation domain edge length L , i.e., $f = \ell/L = 0.1$. The resulting ensemble exhibits randomness of filament orientation, initial stress produced by filament growth, and cross-linkage formation. The ensemble is representative of newly formed actin meshes, with considerable reconfiguration possible by repeated forcing (e.g., as produced by motile behavior).

The microscopic mesh state is characterized by positions, velocities of all monomers, $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t) \in \mathbb{R}^m$, with $m \sim \mathcal{O}(10^5)$ in order to accurately describe a portion of the cytoskeleton of volume $\sim 1 \mu\text{m}^3$, sufficient to determine protrusion force [15]. For a typical human cell volume of $100 \mu\text{m}^3$, the number of microscopic state variables would be $\mathcal{O}(10^7)$. We seek a reduced description of the mesh within a $1 \mu\text{m}^3$ control volume through a linear combination $\mathbf{x} = \mathbf{B}_p \mathbf{y}$, with $\mathbf{B}_p \in \mathbb{R}^{m \times p}$ an orthonormal set of p basis vectors. The reduced linear system obtain by projection of (4) onto $\text{span}(\mathbf{B})$ is

$$\mathbf{D}_p \dot{\mathbf{y}} + \mathbf{K}_p \mathbf{y} = \mathbf{g}, \quad (5)$$

with $\mathbf{D}_p = \mathbf{B}_p^T \mathbf{D} \mathbf{B}_p$, $\mathbf{K}_p = \mathbf{B}_p^T \mathbf{K} \mathbf{B}_p$, and $\mathbf{g} = \mathbf{B}_p^T \mathbf{f}$ the reduced drag, stiffness matrices, and reduced force, respectively.

The drag and stiffness matrices (both at full resolution \mathbf{D} , \mathbf{K} , and reduced resolution \mathbf{D}_p , \mathbf{K}_p) are not known explicitly and cannot be readily evaluated through Taylor series expansions such as $\mathbf{K} = \partial \mathbf{h}(\mathbf{x}_0) / \partial \mathbf{x}$ due to the complex forms of the interaction potentials $U_i(\mathbf{x})$ and $U_i^m(\mathbf{x})$ and the action of the stochastic forcing term ξ . Rather than seeking such an analytical derivation, a data-driven approach is adopted

based upon results obtained from numerical simulation with a forcing term

$$\mathbf{f} = \mathbf{a}x \sin(\omega t) + \boldsymbol{\xi}, \quad (6)$$

that models the solvent stochastic force $\boldsymbol{\xi}$ and a periodic forcing along a single direction as would be produced by an extensional rheometer, $\mathbf{a}x \sin(\omega t)$ (here, x denotes a scalar coordinate). The positional data $\mathbf{x}(t_i)$ at n time steps $t_i = i\delta t$, $\delta t = T/n$, is given by the matrix

$$\mathbf{X} = (\mathbf{x}_1 \quad \dots \quad \mathbf{x}_n) \in \mathbb{R}^{m \times n}. \quad (7)$$

Both the drag and stiffness matrices must be symmetric positive definite due to physical considerations ($\mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0$ gives the elastic deformation energy, $\dot{\mathbf{x}}^T \mathbf{D} \dot{\mathbf{x}} \geq 0$ gives the dissipation energy due to drag of solvent on the monomers), and similar properties are imposed for the reduced matrices through the eigen-decompositions

$$\mathbf{D}_p = \mathbf{V}_p \boldsymbol{\Lambda}_p^v \mathbf{V}_p^T, \mathbf{K}_p = \mathbf{U}_p \boldsymbol{\Lambda}_p^x \mathbf{U}_p^T, \quad (8)$$

in which \mathbf{V}_p and \mathbf{U}_p are orthogonal eigenvector matrices, and the diagonal eigenvalue matrices $\boldsymbol{\Lambda}_p^v$ and $\boldsymbol{\Lambda}_p^x$ have positive components.

The descriptive capability of the reduced model depends crucially on the choice of a basis set \mathbf{B}_p that efficiently captures the possible configurations of the actin mesh in as few modes as possible, $p \ll m$. To this end, the basis set is chosen from the dominant modes of the singular value decomposition (SVD), $\mathbf{X} = \mathbf{B} \boldsymbol{\Sigma} \mathbf{T}^T$. Note that the correlation matrix $\mathbf{C} = \mathbf{X} \mathbf{X}^T$ can be expressed as $\mathbf{C} = \mathbf{B} \boldsymbol{\Sigma} \boldsymbol{\Sigma}^T \mathbf{B}^T$, and the p dominant singular modes $\mathbf{B}_p = (\mathbf{b}_1 \quad \dots \quad \mathbf{b}_p) \in \mathbb{R}^{m \times p}$ correspond to those eigenmodes of the correlation matrix most readily observed during the simulation.

In the reduced model (5), no particular significance is attributed to the components of \mathbf{y} , which are simply the coefficients of the linear combination $\mathbf{x} = \mathbf{B}_p \mathbf{y}$. In some particular cases, it is possible to use straightforward averaging over geometric positions to attach significance to the reduced parameter vector \mathbf{y} , e.g., for one-dimensional elements as in [25]. Such an approach is difficult to justify in higher dimensions though.

Recall that a standard finite element for an elastic body is expressed as $\mathbf{f}^e = \mathbf{K}^e \mathbf{u}^e$, to relate the displacements $\mathbf{u}^e \in \mathbb{R}^{dq}$ at the q control nodes within d dimensions linearly through the element stiffness matrix \mathbf{K}^e to the nodal forces \mathbf{f}^e . The corresponding formulation for a viscoelastic element is

$$\mathbf{D}^e \dot{\mathbf{u}}^e + \mathbf{K}^e \mathbf{u}^e = \mathbf{f}^e. \quad (9)$$

The positions of the control nodes can be chosen arbitrarily but are typically taken to be on the boundary of the finite element in order to invoke continuity of deformation between adjacent elements. In the following, we consider the control nodes at the corners of a three-dimensional right rectangular prism, to obtain the commonly used brick

element, hence $d = 3$, $q = 8$, and the displacements $\mathbf{u}^e(\mathbf{x}) \in \mathbb{R}^d$ within the element are given as

$$\mathbf{u}^e(t, \mathbf{x}) = \sum_{k=1}^q N_k(\mathbf{x}) \mathbf{u}_k^e(t), \quad (10)$$

where $\mathbf{u}_k^e(t) \in \mathbb{R}^d$ are the displacements at node k of the element, and $N_k(\mathbf{x})$ are form functions

$$N_k(\mathbf{x}) = \left\{ \frac{1}{8} (1 \pm x)(1 \pm y)(1 \pm z) \right\}, \quad k = 1, \dots, 8, \quad (11)$$

with x, y, z coordinates from the element center, $\mathbf{x} = (x, y, z)$. The interpolation (10) can be expressed as

$$\mathbf{u}^e(t, \mathbf{x}) = \mathbf{N}^T(\mathbf{x}) \mathbf{u}^e(t), \quad (12)$$

with

$$\mathbf{N}^T(\mathbf{x}) = (N_1(\mathbf{x}) \quad \dots \quad N_q(\mathbf{x})), \mathbf{u}^e(t) = (\mathbf{u}_1^e(t) \quad \dots \quad \mathbf{u}_q^e(t)). \quad (13)$$

We now consider the procedure to link the finite element formulation to the data available from the BD simulation. At time step $t_i = i\delta t$, construct the data set

$$\mathcal{D}_i = \{(\mathbf{x}_j(0), \mathbf{x}_j(t_i) - \mathbf{x}_j(0)), \quad j = 1, \dots, N\} \quad (14)$$

of displacements of all monomers within the mesh $\mathbf{u}_{i,j} \equiv \mathbf{u}(t_i, \mathbf{x}_j(0)) = \mathbf{x}_j(t_i) - \mathbf{x}_j(0)$. A least squares fit of data \mathcal{D}_i to the finite element approximation (12) at time t_i , $\mathbf{u}^e(t_i, \mathbf{x})$, could furnish the nodal values $\mathbf{u}^e(t_i)$, and a subsequent fit to the observed nodal values would provide the element drag and stiffness matrices $\mathbf{D}^e, \mathbf{K}^e$ we seek. However, such a procedure would include all the observed motion, including the stochastic thermal effects. Rather than fitting to \mathcal{D}_i , the nodal values are obtained by fitting onto the data projected onto the dominant p modes contained in the data set

$$\bar{\mathcal{D}}_i = \{(\bar{\mathbf{x}}_j(0), \bar{\mathbf{x}}_j(t_i) - \bar{\mathbf{x}}_j(0)), \quad j = 1, \dots, N\}, \quad (15)$$

with $\bar{\mathbf{x}}_j(t)$ the projection of $\mathbf{x}_j(t)$ onto $\text{span}(\mathbf{B}_p)$, $\bar{\mathbf{x}}_j(t) = \mathbf{B}_p \mathbf{B}_p^T \mathbf{x}_j(t)$. In essence, the dominant modes obtained from the SVD are used as a filter of the observed microscopic motion.

After carrying out the least squares fit to data sets $\bar{\mathcal{D}}_i$, $i = 1, \dots, n$, the nodal displacements $\mathbf{u}^e(t_i)$ are known, and the nodal velocities can be approximated by, e.g., finite difference approximations

$$\mathbf{v}^e(t_i) \equiv \dot{\mathbf{u}}^e(t_i) \cong \frac{\mathbf{u}^e(t_{i+1}) - \mathbf{u}^e(t_{i-1})}{2\delta t}. \quad (16)$$

As in the case of the full drag, stiffness matrices, similar physical constraints are imposed onto the finite element matrices

$$\begin{aligned} \mathbf{D}^e &= \mathbf{V}^e \Lambda_v^e (\mathbf{V}^e)^T, \\ \mathbf{K}^e &= \mathbf{U}^e \Lambda_x^e (\mathbf{U}^e)^T, \end{aligned} \quad (17)$$

with $\mathbf{V}^e, \mathbf{U}^e \in \mathbb{R}^{(dq) \times (dq)}$ orthogonal matrices and components of diagonal matrices Λ_v^e, Λ_x^e constrained to be positive.

The columns of \mathbf{V}^e and \mathbf{U}^e describe the velocity and positional modes of the finite element. In standard finite element analysis of an elastic body, such modes would result from hypotheses on the deformation behavior. Typically, stretch, shear, twist, and bending of the finite element would be of interest, assumed to be independent of another (i.e., orthogonal modes), and used to construct \mathbf{V}^e and \mathbf{U}^e . The components of Λ_v^e and Λ_x^e would result from, say, a Lagrangian formulation of the equations of motion. Such an approach is reasonable for a homogeneous material, but the deformation modes of an actin mesh are likely to be strongly dependent on mesh configuration and can be expected to always couple the intuitively derived stretch, shear, twist, and bending modes. Rather than impose assumed deformation modes, we seek to construct these from an ensemble of actin mesh configurations.

Consider s instances of actin mesh configurations $\mathbf{X}^1, \dots, \mathbf{X}^s \in \mathbb{R}^{m \times n}$ obtained from s distinct BD simulations (1), from which the basis sets $\mathbf{B}_p^1, \dots, \mathbf{B}_p^s \in \mathbb{R}^{m \times p}$ are obtained through SVDs. The ensemble average basis set

$$\tilde{\mathbf{B}}_p = \frac{1}{s} \sum_{l=1}^s \mathbf{B}_p^l \quad (18)$$

captures both variability from thermal forcing and that due to mesh configuration through different initial states $\mathbf{x}^1(0), \dots, \mathbf{x}^s(0)$. The k^{th} column of $\mathbf{U}^e \in \mathbb{R}^{(dq) \times (dq)}$ is denoted by $\mathbf{u}_k^e \in \mathbb{R}^{(dq)}$ can be interpreted as the coefficient set needed to reconstruct an approximation of the microscopic mesh displacement $\mathbf{u}_k(\mathbf{x})$ through the finite element interpolation (12) as

$$\mathbf{u}_k(\mathbf{x}) = \mathbf{x} - \mathbf{x}(0) \cong \begin{pmatrix} \mathbf{N}^T(\mathbf{x}_1) \mathbf{u}_k^e \\ \vdots \\ \mathbf{N}^T(\mathbf{x}_N) \mathbf{u}_k^e \end{pmatrix} \equiv \mathbf{w}(\mathbf{x}, \mathbf{u}_k^e). \quad (19)$$

A link can now be made to the available data by choosing $p = dq$ and computing the solutions \mathbf{z}_k^e , for $k = 1, \dots, p$ of the p least squares problems

$$\min_{\mathbf{u}_k^e} \left\| \tilde{\mathbf{b}}_k - \mathbf{x}(0) - \mathbf{w}(\mathbf{x}, \mathbf{z}_k^e) \right\|. \quad (20)$$

The vectors \mathbf{z}_k^e are not necessarily orthogonal as desired for the columns of \mathbf{U}^e , so a QR -decomposition $\mathbf{U}^e \mathbf{R} = \mathbf{Z}^e \equiv (\mathbf{z}_1^e \dots \mathbf{z}_p^e)$ is computed to obtain \mathbf{U}^e .

Note that the above procedure homogenizes both variability in mesh configuration and that due to stochastic effects to obtain data-driven deformation modes. An analogous procedure is applied to the velocity data $\dot{\mathbf{X}}$ to obtain the velocity modes \mathbf{V}^e . For a purely elastic system, the velocity modes would be identical to the deformation modes, but for viscoelastic systems, the modes can differ due to phase differences and history dependence (note that even though each monomer is affected by the same type of thermal stochastic forcing ξ , the interaction potentials can induce correlation between the stochastic modes of the overall system).

Once the reduced modes \mathbf{U}^e and \mathbf{V}^e are determined, the remaining task to completely define the finite element model is to compute the diagonal matrices Λ_x^e and Λ_v^e by a least-squares fit over the time history $\{\mathbf{u}^e(t_i), \dot{\mathbf{u}}^e(t_i)\}$ determined by fitting to data sets $\tilde{\mathcal{D}}_i$, i.e., solving the problem

$$\min_{\Lambda_x^e, \Lambda_v^e} \sum_{i=1}^n \left\| \mathbf{V}^e \Lambda_v^e (\mathbf{V}^e)^T \dot{\mathbf{u}}^e(t_i) + \mathbf{U}^e \Lambda_x^e (\mathbf{U}^e)^T \mathbf{u}^e(t_i) - \mathbf{f}^e(t_i) \right\|. \quad (21)$$

3. Results

An initial test of the above data-driven finite element construction procedure is carried out for data from seven network instances forced at nondimensional periods of $T = 1.57, 3.14$ (see [15] for details on reference units used to obtain nondimensional quantities). The goal here is to carry out basic verification of the model reduction procedure rather than obtain a fully realistic description of the biomechanics of F-actin networks. The networks contain $N = 1706$ monomers at $n = 2000$ evenly spaced time steps during the period T . A typical sequence of monomer displacement vectors is shown in Figure 1 and is included here mainly to highlight the difficulty in ascertaining dominant deformation types from such data. Though the forcing (6) is applied along a single coordinate direction, the random nature of filamentary orientation and cross-linkage leads to a complex response of the structure.

A stress-strain curve along the forcing direction may be extracted from the microscopic data and compared to that obtained by integrating the finite element model (9). The results are presented in Figure 2. The microscopic model shows considerable variation and hysteresis over the period T . During the first quarter period $[0, T/4]$ corresponding to rising extensional stress applied to the system, the filament network on average shows rising strain but interspersed with strain release events that probably correspond to network reconfiguration. During the next half period $[T/4, 3T/4]$ of decreasing extensional stress, strain release occurs but with clear plastic behavior due to the viscous terms in the system, and the final quarter period $[3T/4, T/2]$ exhibits a different average slope indicating strain hardening.

No claim is made within these preliminary results of accurate capturing of true biological F-actin meshes, but the reduced finite element model constructed by seeking periodic solutions of (9) $\mathbf{u}^e(t) = \mathbf{a}_0 + \sum_{k=1}^8 (\mathbf{a}_k \cos(2\pi kt/T) +$

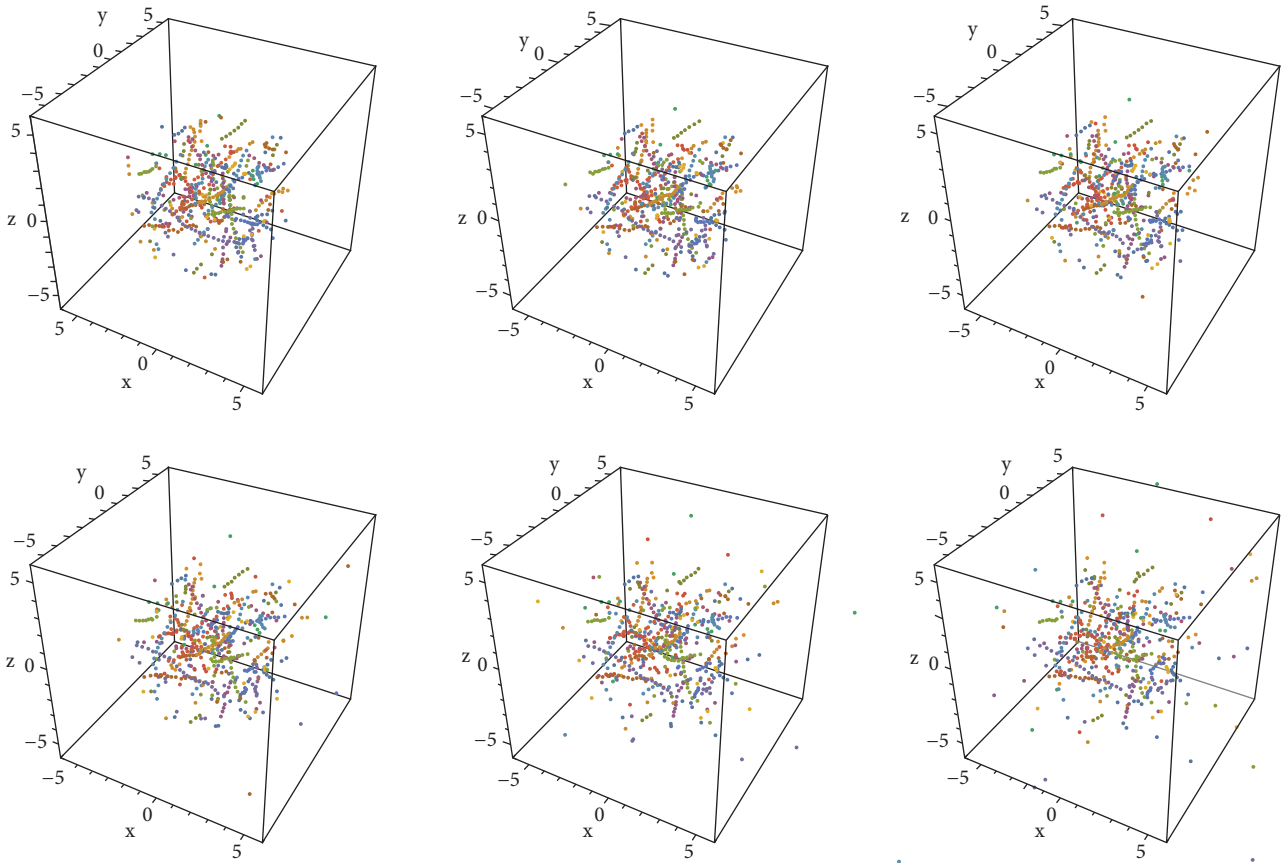


FIGURE 1: Sequence of actin monomer positions at six equidistant times during a period of the forcing. Notice that some free actin monomers exhibit large displacements, and the overall response to forcing along a single coordinate direction is difficult to ascertain from this data.

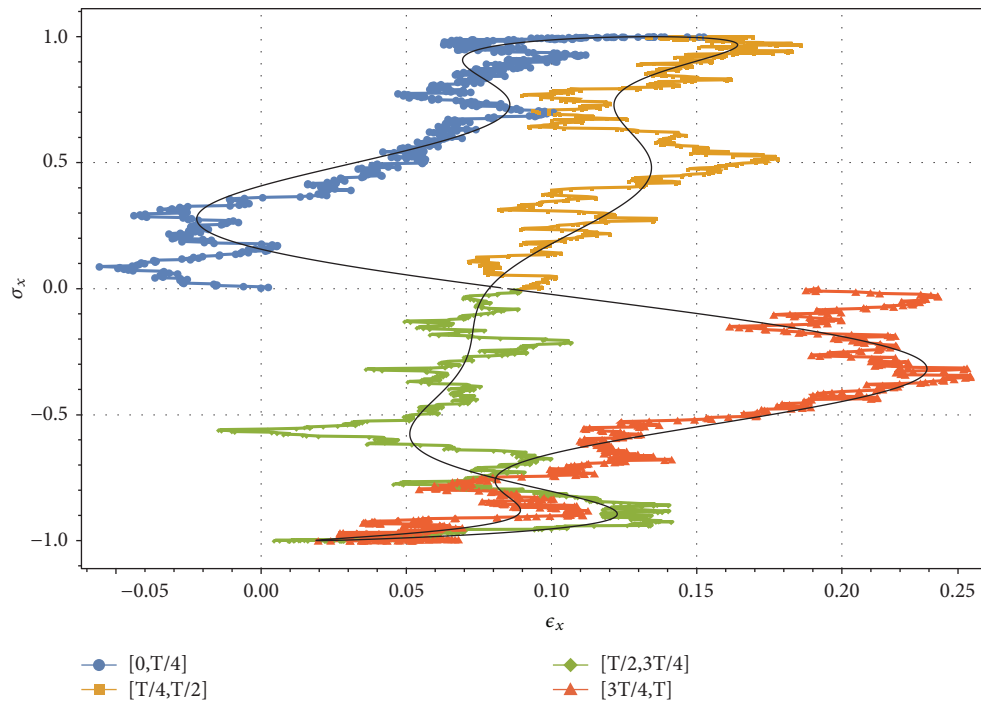


FIGURE 2: Comparison of strain-stress curve of microscopic model (plot markers, distinct for each quarter period of the forcing period T) and the strain stress curve resulting from the finite element model (9).

$\mathbf{b}_k \sin(2\pi kt/T)$ captures the main features of the microscopic simulation to within a relative error of 3.3% for the stress-strain curve. Note that the microscopic data exhibit initial transient behavior, not the long-term periodic behavior that results from repeated application of the oscillatory extensional force (6), and this is observed in the behavior at $t \cong 0$ and $t \cong T$.

4. Discussion

A general procedure has been introduced to define a finite element model from data obtained by Brownian dynamics simulation of an F-actin network. Though the presentation used data from numerical simulation and exemplified the procedure using an eight-node cuboid finite element, the approach can readily be applied to obtain different types of finite element approximation or use data from experiments. Of particular interest is the extraction of essential behavior from the available data without any assumption on deformation modes or constitutive relationships within the material. For the case of the F-actin network considered here, the derived finite element approximation captures the stress-strain relationship from the microscopic model with a relative error of $\epsilon_r \leq 0.04$, at negligible computational cost by comparison to Brownian dynamics simulations. Multiple application of the procedure can be used to model nonlinear behavior or parametrized to model active systems concurrently with biochemical reaction models.

At this point, the reduced finite element model is deterministic, capturing the average behavior of the filamentary network. The procedure however forms the basis of establishing a stochastic finite element approximation through recognition of the close relationship between the singular value decomposition (SVD) and the Karhunen-Loeve expansion. In this work, only the singular modes from the SVD were used in constructing a deterministic finite element approximation. Such modes correspond to the eigenmodes in a Karhunen-Loeve expansion, while the singular values give the mode variances. Use of the variances would allow formulation of a stochastic finite element, the subject of upcoming work.

Data Availability

Simulation data and codes are available and may be directly requested from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

BA carried out numerical simulations of the F-actin network, and SM developed the model reduction procedure to define a finite element model.

Acknowledgments

This work was supported by the NSF (CMMI-1068918, DMS-1361375).

References

- [1] L. Blanchoin, R. Boujemaa-Paterski, C. Sykes, and J. Plastino, "Actin dynamics, architecture, and mechanics in cell motility," *Physiological Reviews*, vol. 94, no. 1, pp. 235–263, 2014.
- [2] A. S. Abhilash, B. M. Baker, B. Trappmann, C. S. Chen, and V. B. Shenoy, "Remodeling of fibrous extracellular matrices by contractile cells: predictions from discrete fiber network simulations," *Biophysical Journal*, vol. 107, no. 8, pp. 1829–1840, 2014.
- [3] J. Stricker, T. Falzone, and M. L. Gardel, "Mechanics of the F-actin cytoskeleton," *Journal of Biomechanics*, vol. 43, no. 1, pp. 9–14, 2010.
- [4] M. L. Gardel, J. H. Shin, F. MacKintosh, L. Mahadevan, P. Matsudaira, and D. A. Weitz, "Elastic behavior of cross-linked and bundled actin networks," *Science*, vol. 304, no. 5675, pp. 1301–1305, 2004.
- [5] C. Storm, J. J. Pastore, F. C. MacKintosh, T. C. Lubensky, and P. A. Janmey, "Nonlinear elasticity in biological gels," *Nature*, vol. 435, no. 7039, pp. 191–194, 2005.
- [6] C. P. Broedersz and F. C. MacKintosh, "Modeling semiflexible polymer networks," *Reviews of Modern Physics*, vol. 86, no. 3, pp. 995–1036, 2014.
- [7] F. Meng and E. M. Terentjev, "Theory of semiflexible filaments and networks," *Polymers*, vol. 9, no. 12, 2017.
- [8] M. Klann and H. Koepl, "Spatial simulations in systems biology: from molecules to cells," *International Journal of Molecular Sciences*, vol. 13, no. 6, pp. 7798–7827, 2012.
- [9] T. Kim, W. Hwang, H. Lee, and R. D. Kamm, "Computational analysis of viscoelastic properties of crosslinked actin networks," *PLoS Computational Biology*, vol. 5, no. 7, article e1000439, 2009.
- [10] G. Zagar, P. R. Onck, and E. van der Giessen, "Two fundamental mechanisms govern the stiffening of cross-linked networks," *Biophysical Journal*, vol. 108, no. 6, pp. 1470–1479, 2015.
- [11] O. Lieleg, K. M. Schmoller, M. M. A. E. Claessens, and A. R. Bausch, "Cytoskeletal polymer networks: viscoelastic properties are determined by the microscopic interaction potential of cross-links," *Biophysical Journal*, vol. 96, no. 11, pp. 4725–4732, 2009.
- [12] M. H. Jensen, E. J. Morris, and D. A. Weitz, "Mechanics and dynamics of reconstituted cytoskeletal systems," *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, vol. 1853, no. 11, pp. 3038–3042, 2015.
- [13] C. J. Cyron, K. W. Müller, A. R. Bausch, and W. A. Wall, "Micromechanical simulations of biopolymer networks with finite elements," *Journal of Computational Physics*, vol. 244, pp. 236–251, 2013.
- [14] J. Zhang, X. Zhao, Z. Suo, and H. Jiang, "A finite element method for transient analysis of concurrent large deformation and mass transport in gels," *Journal of Applied Physics*, vol. 105, no. 9, 2009.
- [15] Y. Inoue, T. Deji, and T. Adachi, "Brownian dynamics simulation study on force-velocity relation in actin-based membrane

- protrusion,” *Computational Particle Mechanics*, vol. 2, no. 4, pp. 329–337, 2015.
- [16] G. G. Borisy and T. M. Svitkina, “Actin machinery: pushing the envelope,” *Current Opinion in Cell Biology*, vol. 12, no. 1, pp. 104–112, 2000.
- [17] K. Keren, Z. Pincus, G. M. Allen et al., “Mechanism of shape determination in motile cells,” *Nature*, vol. 453, no. 7194, pp. 475–480, 2008.
- [18] A. Mogilner and G. Oster, “Cell motility driven by actin polymerization,” *Biophysical Journal*, vol. 71, no. 6, pp. 3030–3045, 1996.
- [19] A. Mogilner and G. Oster, “Force generation by actin polymerization ii: the elastic ratchet and tethered filaments,” *Biophysical Journal*, vol. 84, no. 3, pp. 1591–1605, 2003.
- [20] A. Mogilner and B. Rubinstein, “The physics of filopodial protrusion,” *Biophysical Journal*, vol. 89, no. 2, pp. 782–795, 2005.
- [21] J. Gernandt, G. Frenning, W. Richtering, and P. Hansson, “A model describing the internal structure of core/shell hydrogels,” *Soft Matter*, vol. 7, no. 21, pp. 10327–10338, 2011.
- [22] J. M. Holzwarth and P. X. Ma, “Biomimetic nanofibrous scaffolds for bone tissue engineering,” *Biomaterials*, vol. 32, no. 36, pp. 9622–9629, 2011.
- [23] P. X. Ma and R. Y. Zhang, “Synthetic nano-scale fibrous extracellular matrix,” *Journal of Biomedical Materials Research*, vol. 46, no. 1, pp. 60–72, 1999.
- [24] T. A. Ulrich, A. Jain, K. Tanner, J. L. MacKay, and S. Kumar, “Probing cellular mechanobiology in three-dimensional culture with collagen–agarose matrices,” *Biomaterials*, vol. 31, no. 7, pp. 1875–1884, 2010.
- [25] Y. Feng and S. Mitran, “Data-driven reduced-order model of microtubule mechanics,” *Cytoskeleton*, vol. 75, no. 2, pp. 45–60, 2018.

Research Article

A Data-Driven Parameter Adaptive Clustering Algorithm Based on Density Peak

Tao Du , Shouning Qu , and Qin Wang 

School of Information Science and Engineering, University of Jinan, No. 336, West Road of Nan Xinzhuang, Jinan 250022, Shandong, China

Correspondence should be addressed to Tao Du; ise_dut@ujn.edu.cn

Received 31 May 2018; Accepted 6 August 2018; Published 21 October 2018

Academic Editor: Rafael Gómez-Bombarelli

Copyright © 2018 Tao Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Clustering is an important unsupervised machine learning method which can efficiently partition points without training data set. However, most of the existing clustering algorithms need to set parameters artificially, and the results of clustering are much influenced by these parameters, so optimizing clustering parameters is a key factor of improving clustering performance. In this paper, we propose a parameter adaptive clustering algorithm DDPA-DP which is based on density-peak algorithm. In DDPA-DP, all parameters can be adaptively adjusted based on the data-driven thought, and then the accuracy of clustering is highly improved, and the time complexity is not increased obviously. To prove the performance of DDPA-DP, a series of experiments are designed with some artificial data sets and a real application data set, and the clustering results of DDPA-DP are compared with some typical algorithms by these experiments. Based on these results, the accuracy of DDPA-DP has obvious advantage of all, and its time complexity is close to classical DP-Clust.

1. Introduction

Clustering is one of the most important methods in machine learning, and by clustering, data points are partitioned to several groups [1], and the ones in the same group are much similar, and points in different groups are much different [2–4]. Clustering algorithm can deal with data points without any labelled samples, so it is much fit for the fast-changing environment, in which the samples are hardly obtained [5]. Nowadays, with the development of big data, clustering has been more and more applied in Internet of things, environment monitoring, image processing, etc. [6].

There have been more and more researches focused in designing high efficient clustering algorithm, and these researches can be divided to four kinds: the partition-based methods, such as K-means [7] and K-medoids [8]; the hierarchy-based methods, such as BIRCH [9], ROCK [10], and Chameleon [11]; the density-based methods, such as DBSCAN [12] and OPTICS [13]; and grid-based methods, such as STING [14] and CLIQUE [15]. In classical partition-based algorithms, the number of clusters should be

artificially defined before clustering, which much restricts the flexibility of clustering application, and they are not able to efficiently cluster the nonhypersphere data set [1]. In classical hierarchy-based algorithms, the threshold of merging microclusters or dividing macroclusters is the key parameter of clustering, and it is also set artificially before clustering [16], and these algorithms' time complexity is relatively large. In classical grid-based algorithms, grid granularity is the key parameter, and the clustering accuracy will be poor if it is set too large, otherwise, the time complexity will be much increased if it is set too little [17]. Density-based clustering algorithms can cluster arbitrary shapes of data sets and the clustering results are not influenced by noise points, so density-based algorithms have been the focus of clustering researches, and there have been many new algorithms proposed [18–20]. Density peak-based clustering (DP-Clust) is one of the important algorithms of these researches, and on the basis of the advantages of density based algorithms, DP-Clust improves the efficiency of clustering by detecting centers, borders, and outliers from all data points [21]. However, as

other density-based algorithms, DP-Clust needs to set the local field's radius of every point to accomplish clustering, and the thresholds of detecting centers and outliers are also set in advance, and then the performance is not good at dealing with sparse distribution data set.

Based on the above analysis, it can be seen that the artificial setting of clustering parameters has been the key factor of influencing the performance of clustering, so nowadays, some researchers have focused on optimizing parameters to improve clustering efficiencies: FEAC can adapt the number of clusters which was proposed by Silva to get rid of the defect K-means [22], however the complexities of time and memory are too large and it cannot efficiently cluster arbitrary shapes of data sets; Hou proposed a parameter independent hierarchy based algorithm named DSets-histeq [23], in which microclusters are merged according to the theory of dominant set, and it can automatically adjust parameters by establishing similarity matrices of every pair of microclusters, so the complexity of clustering is much increased; Myhre proposed a grid-based algorithm named KNN-MS, in which data points are partitioned to K grids, and by mode seeking theory, all grids would be adjusted and the result are not influenced by the value of K [24], but it is much influenced by noise points. And there are many other algorithms proposed to reduce the influence of the parameter's initial value; however, these ones have defects in dealing with arbitrary shapes of data sets or the efficiency in clustering. Then to use the advantages of high efficiency in clustering arbitrary data sets and relative simple clustering procedure of DP-Clust, many density peak-based algorithms are proposed to realize self-adapting parameters to improve clustering performance; however, these researches have more or less artificial factors when setting clustering parameters, and they cannot realize fully data-driven parameter adaptive clustering algorithm based on density peak. To improve the independence of parameters when clustering, we propose a fully data-driven parameter adaptive clustering algorithm based on density peak (DDPA-DP). In DDPA-DP, all parameters can be updated by the distribution of data points, and the procedure of adapting parameters is simple to be accomplished to reduce the time complexity of whole algorithm. The rest of this paper is organized as follows: in section two, the basic thought of DP-Clust is introduced, and related researches are analyzed; in section three, the thought of DDPA-DP is proposed, and the detail of this algorithm is designed; in section four, a series of experiments are simulated, and the other three algorithms are compared with DDPA-DP; and at last, the contribution of this paper is concluded.

2. Related Work

2.1. The Introduction of Density Peak Clustering. In 2014, Rodriguez proposed a density-based clustering algorithm named DP-Clust, and the basic thought of DP-Clust is that the centers of all clusters should be located at the peak of local density changing curve, and the borders will be located at the neighborhoods of centers, and outliers will be far away from

high-density area [21]. To detect centers, borders, and outliers, two conceptions are defined in DP-Clust:

Definition 1. Local density is an attribute to measure the density station of point i by computing the influence of other points in point i neighborhood to point i , and it can be computed as (1) or as (2).

$$\rho_i = \sum X \times (d(i, j) - r), \quad X = \begin{cases} 1, & d(i, j) > r, \\ 0, & d(i, j) < r, \end{cases} \quad (1)$$

$$\rho_i = \sum \exp\left(\frac{d(i, j)^2}{r^2}\right). \quad (2)$$

In (1) and (2), r is a cutoff distance, and the radius of point i 's neighborhood is r and the center is i . Then $d(i, j)$ is the distance from i to its neighbor j which is located in i 's neighborhood. By (1) and (2), just the points in i 's neighborhood can influence its local density. After all points' local densities are obtained, a list L will be established, and in L all points will be rearranged with the descending order of their local densities as $\{\rho_{q1}, \rho_{q2}, \rho_{q3}, \dots, \rho_{qn}\}$.

Definition 2. The distance from the nearest neighbor with larger local density than i is defined as (3), and this is an attribute to measure the point whether be located in the center of a high-density field.

$$\delta_{qi} = \begin{cases} \min_{j < i} \{d_{qi,qj}\}, & i \geq 2 \\ \max_{j \geq 2} \{d_{qi,qj}\}, & i = 1 \end{cases}. \quad (3)$$

According to (3), if point i is the first one in list L, the value of δ_{qi} is set as the distance to the farthest point from i ; otherwise, the value of δ_{qi} is set as the distance from i to the nearest point whose position in L is in front of i .

After obtaining the ρ and δ of every point, the one has both larger ρ and δ can be detected as centers, because larger ρ means this point located in a high-density area, and larger δ means there are not any points in the same high-density area with larger ρ than it, and then it can be seemed as the center of this area. Otherwise, if the point has less ρ and larger δ , it can be detected as outlier, because less ρ means this point is located in a sparse area, and meanwhile, larger δ means this point is far away any high-density area, and then it can be seemed as be out of all clusters. At last, all remaining points can be detected as borders, and these points have larger ρ and less δ , which means every border is located in a high-density area, but there is at least one point in the same area located nearer to the center. After all points' roles are being obtained, every border will join the nearest center to format cluster.

Because just local density instead of global density needs to be computed, DP-Clust has obvious advantage in clustering nonuniform density fields comparing to DBSCAN, and its clustering procedure is simple to be deployed in

application. Now, DP-Clust has dropped much attention, and many density peak-based clustering algorithms have been proposed [25–27]; however, same as DP-Clust, these algorithms should set three main thresholds as the radius of local field r , the standards of larger or less of local density, and the standards of larger or less of the nearest distance to neighbor with larger local density, and these settings restrict the clustering performance especially in sparse and changeable environments.

2.2. Existing Researches of Parameter Optimized Density Peak-Based Clustering. To reduce the influence of initial setting of parameters of DP-Clust, there are two problems to be resolved: one is how to determine the thresholds of local density ρ and the distance to the nearest neighbor with larger local density δ ; the other is how to select optimizing radius of local field r .

In ref [28], Chen and He proposed an algorithm named ACC-FSFD, in which a curve fitting method is adopted to automatically find the points with both larger ρ and δ to determine centers. ACC-FSFD designed a variable $\gamma = \rho * \delta$, and centers will be detected by finding the points with obvious larger γ than the value predicting by the curve fitting function. Although ACC-FSFD can automatically obtain centers, it did not take outliers into account, which leads the accuracy is much influenced by noise. In ref [29], Saki and Kehtarnavaz used a histogram to reflect the distribution of all points' ρ and δ , and centers and outliers will be detected by data-driven method; however, this algorithm time complexity is large, because it takes too many calculations when establishing histogram. In ref [30], Xu et al. proposed a FNLT algorithm, in which two different linear-regression analysis functions are established to, respectively, detect centers and outliers, and clusters will be stored as Leading Tree and centers as Fat Nodes of these trees, and by merging trees to optimize the distribution of clusters. Although FNLT can detect points' roles by data driven, the clustering procedure is too complex to be deployed because of the complex structure of FNLT, and the linear-regression analysis method is less accurate in predicting the change tendency of ρ and δ .

Besides the defects of setting thresholds of centers and outliers, the algorithms mentioned above adopt fixed and preset radius of local field to accomplish clustering, which much restricts the performance in complex environments. Nowadays, there are two kinds of effective algorithms that focus on optimizing the local field radius of density peak clustering: one adopts K nearest neighbors-based method to divide the local field instead of by radius; the other directly optimizes the local field's radius to reduce the affection of initial setting.

DPC-KNN is a classical KNN-based algorithm [26], and in DPC-KNN, the points' local density is computed as (4).

$$\rho_i = \exp \left(-\frac{1}{K} \sum_{j \in KNN_i} d_{ij}^2 \right). \quad (4)$$

FKNN-DPC [31] is another KNN-based one, and the local density in FKNN-DPC is computed as (5).

$$\rho_i = \sum_{j \in KNN_i} \exp(-d_{ij}). \quad (5)$$

Comparing (1) and (2) with (4) and (5), in the KNN-based algorithms, a point will obtain its local density by computing the distances to its K neighbors, and the parameter K needs to be input in advance. The advantage of KNN-based method is that clustering complexity will be much less than DP-Clust if all points' KNN have been known. However, in most application environments, the operation of obtaining KNN of every point will be so hard that the performance is not obviously improved. Liu et al. proposed an adaptive KNN-based algorithm ADPC-KNN [32], and in this algorithm, the local density is computed by combination of DP-Clust and DPC-KNN as (1), in which r is deduced by K as (6) and (7), and the value of K can be adjusted by evaluating the distribution of clusters.

$$r = \mu^K + \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\delta_i^K - \mu^K)^2}, \quad (6)$$

$$\mu^K = \frac{1}{N} \sum_{i=1}^N \delta_i^K, \quad (7)$$

where N is the number of all points, and $\delta_i^K = \max_{j \in KNN_i} d_{ij}$ is the distance from i to its K th nearest neighbor. Based on (6), ADPC-KNN can optimize the value of r by all data points; however, the calculation complexity is much increased. Besides the clustering efficiency, the performances of KNN-based algorithms are relatively poor in nonuniform fields, because in these fields, density in a point's KNN will be much different with others, which leads to the centers in sparse area cannot be well detected. And the value of K will influence the result of clustering, which does not well satisfy the demand of parameter independence.

In ref [33], a DP-Clust-based algorithm named DCore was proposed by Chen et al. DCore uses a concept of density core to find high-density fields and to determine centers and borders, in which the clusters in sparse area can be detected by mean shift thought. In Dcore, data-driven thought is used to adjust the clusters' distribution; however, the value of r is fixed, and the threshold of determine centers is artificially set which restrict the DCore's performance in nonuniform fields. Based on Dcore, DCNaN was proposed by Xie et al. [34], and in DCNaN, every point will compute its local field's radius by (8).

$$r_i = \frac{\sum_{j \in NaN(i)} d(i, j)}{b(i)}. \quad (8)$$

In (8), $NaN(i)$ is the natural neighbors' set of point i , and $b(i)$ is the number of natural neighbors of point i , and the concept of natural neighbors was introduced in ref [ccc]

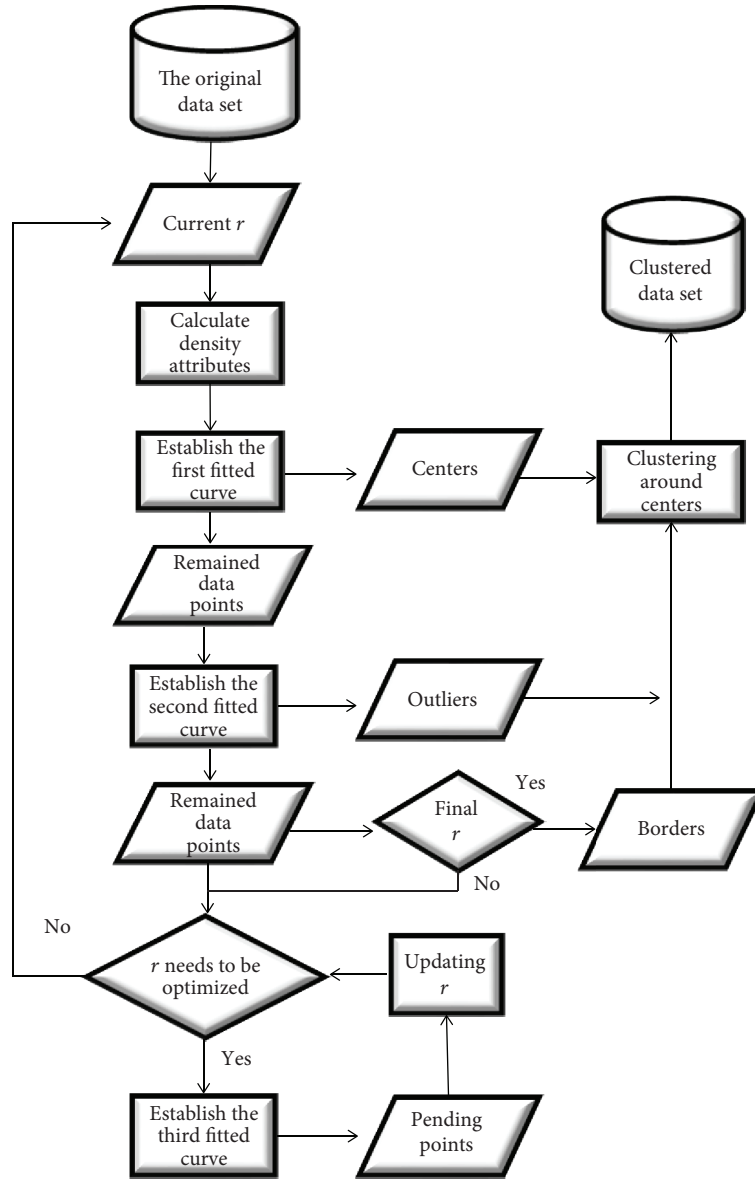


FIGURE 1: The flow of DDPA-DP.

and [ddd]. Then, a list of sorted scanning radiuses r is established, and by computing the variation of adjacent r in this list, centers and outliers are detected. In this algorithm, the data-driven method in dynamical adjusting local field's radius improves the clustering performance in sparse area; however, in DCNaN, the thresholds of judging natural neighbors, centers, and outliers are preset and fixed, and the procedure of adjusting the local field's radius needs too many iterative computations to much increase the complexity of clustering.

Based on the application of clustering, it can be concluded that there are three key problems that should be resolved when designing clustering algorithm: the accuracy in clustering arbitrary data set, the parameter independence when clustering, and the complexity of time and memory. However, based on the above analysis, existing researches have much or less defects so that these problems are not

well resolved, and now these problems have been the major obstacles of restricting the clustering application.

3. The Design of DDPA-DP

To obtain the target of improving clustering performance, we propose a fully data-driven parameter adaptive clustering algorithm based on density peak (DDPA-DP), and in DDPA-DP, the parameter of the local field's radius r can be dynamically adjusted, and the thresholds of detecting points' roles are determined by data distribution, and the complexity of this algorithm is also better than classical density-based ones. In DDPA-DP, there are three steps: density attributes are computed by initial value of r , and then points' roles are automatically detected, and a self-adaptive procedure will be called to optimize r . The flow of DDPA-DP is shown in Figure 1.

In the first step of DDPA-DP, r will be set an initial value, and then all points' local density ρ and the distance to the nearest higher density neighbor δ will be computed with the current value of r ; secondly, according to the current points' values of ρ and δ , a series of fitting curves will be established, and the points whose ρ and δ are obviously different to most of the others will be detected, and their roles will be determined based their ρ and δ ; at last, by the distribution of points' roles, the value of r will be evaluated and optimized. Repeat these three steps until r is convergent.

3.1. Automatically Detecting Points' Roles. According to the thought of DP-Clust, we propose a series of fitted curves to predict the combination value of ρ and δ , and based on the distribution of the difference between predicting value and real value of every point, a point's ρ and δ are larger or less one among all points can be automatically determined, and then by the density attributes of all points, their roles can be detected. Meanwhile a new kind of points named "pending point" is defined as Definition 3. To better illustrate the algorithm, a simple model as Figure 2 is established.

Definition 3. When a point's ρ is less than most points and meanwhile its δ is less than most points too, it is hard to determine that this point belongs to borders or outliers, so we call these points "pending point." This role does exist in data points, but existing algorithms do not research it.

Assuming $D\{d_1, d_2, d_3, \dots, d_N\}$ is the target data set with N points, and every point has n attributes. Centers' ρ and δ are both larger than most of the other points, so a variable $\gamma = \delta \times \rho$ is defined to establish the fitted curve as (9).

$$\gamma = a_0 + a_1 \times I_c + a_2 \times I_c^2 + \dots + a_N \times I_c^N. \quad (9)$$

In (9), $I_c \{i_{c1}, i_{c2}, \dots, i_{cM}\}$ is the index of N data points, which is used to act as independent variable, and $\{a_0, a_1, a_2, \dots, a_N\}$ are the coefficients of fitted curve. By this curve, if a data point's index is known, the value of γ can be predicted. The difference between the real value of γ^* and its predicted value γ is $\Delta\gamma$, and then both the mean and the variance of $\Delta\gamma$ can be obtained. The frequency histogram of the distribution of $\Delta\gamma$ is shown in Figure 3. In Figure 3, ϵ is the mean of $\Delta\gamma$ and its value is 0, which means that the predicting values of γ of most points are very close to their real value, and σ is the variance of $\Delta\gamma$. From Figure 3, most of the points are distributed in the value range $\{-\sigma \leq \Delta\gamma \leq \sigma\}$. When a point's $\Delta\gamma > \sigma$, it means that this point has larger γ than most of the other points, and it can be seemed as candidate center, and the corresponding relations are shown in Figure 4. Based on this procedure, the thresholds of judging centers are determined by the distribution of points and they need not be artificially set, which reflects the advantages of data-driven thought.

After detecting centers, outliers should be detected from remained points. So a variable $\gamma' = \delta \div \rho$ is defined to find

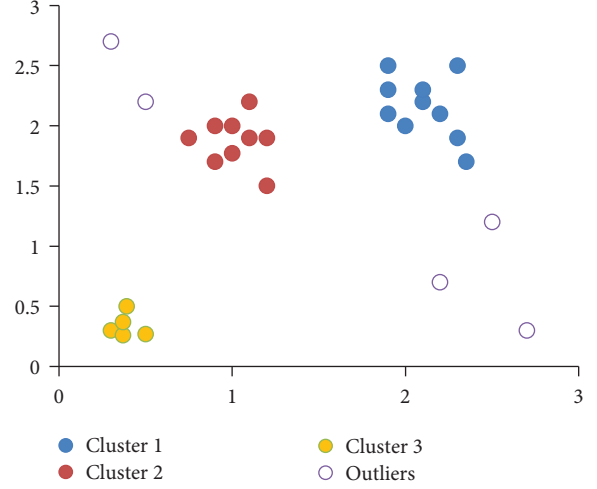


FIGURE 2: A simple model of clusters.

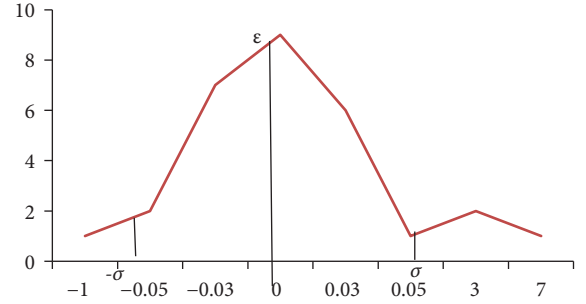


FIGURE 3: The distribution of $\Delta\gamma$.

the points with larger δ and less ρ , and a fitted curve as (10) is established to predict the value of γ' .

$$\gamma' = b_0 + b_1 \times I_c + b_2 \times I_c^2 + \dots + b_m \times I_c^m. \quad (10)$$

By (10), $\Delta\gamma'$ is obtained which is the difference between γ'^* and γ' , and the frequency histogram of $\Delta\gamma'$ is shown in Figure 5, in which most of the points' $\Delta\gamma'$ are distributed in the value range $\{\Delta\gamma' \leq \sigma\}$. Then, outliers can be automatically detected by finding the points whose $\Delta\gamma'$ is larger than σ , and the corresponding relations between outliers and their values of $\Delta\gamma'$ are shown in Figure 6.

As the operations of detecting centers and outliers, pending points can be detected by finding the points with less δ and less ρ , so a variable $\omega = \delta \div \rho$ is defined, and a fitted curve as (11) is established to predict the value of ω . When the difference between real value and predicted value of a point is larger than the variance, it can be seemed as pending points, which is shown in Figure 7.

$$\omega = c_0 + c_1 \times I_c + c_2 \times I_c^2 + \dots + c_m \times I_c^m. \quad (11)$$

After centers, outliers and pending points are detected, remained points can be seemed as borders of clusters, and

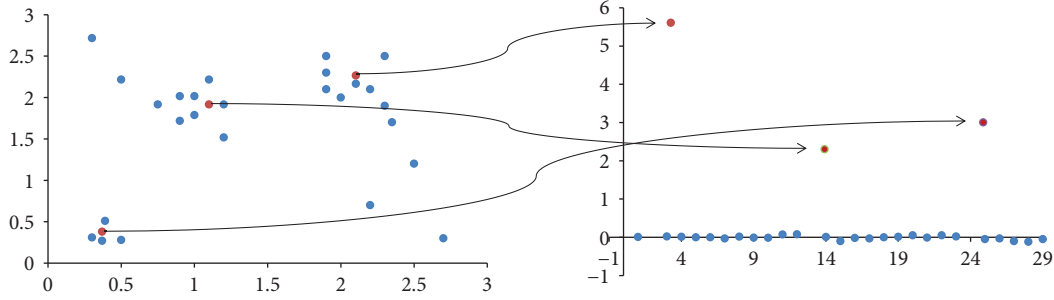
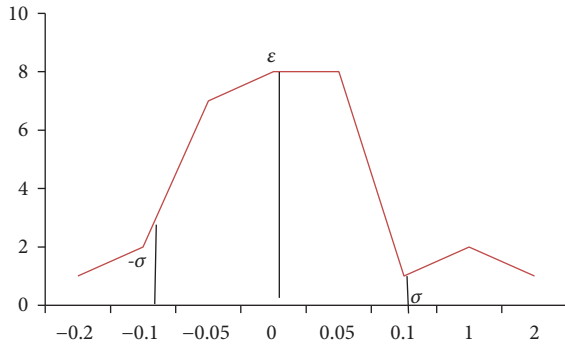


FIGURE 4: The corresponding points of centers.

FIGURE 5: The distribution of $\Delta y'$.

then every border will join the nearest center and form the cluster, and the result is shown in Figure 8.

3.2. Optimizing the Radius of Local Field. In Figure 8, there are two pending points, and these points have less ρ which means they are located in sparse area; however they have less δ which means every pending point has at least one neighbor with larger local density than it. As in Figure 8, the pending points' local field connects a relative dense area and a relative sparse area, and it is hard to determine the role of pending point in cluster. So the distribution of pending point means that the radius of points' local field is not well set, and then we propose a self-adaptive method to optimize the value of local field radius as (12).

$$\begin{aligned} \Delta r &= (r_{i-1} - r_{i-2}) \times (-1)^x \div p_{i-1} \times e^{-|p_{i-2} - p_{i-1}|/\rho_{n-1}}, i \geq 2, \\ \Delta r &= (r_{i-1} - r_{i-2}) \div (\rho_{i-1} - \rho_{i-2}), \quad \rho_{i-1} = \rho_{i-2}. \end{aligned} \quad (12)$$

In (12), Δr is the adjustment quantity of local field radius, and r_{i-1} is the current value of radius which is used to detect points' roles in the last round of computing, and r_{i-2} is the last value of radius; p_{i-1} is the number of pending points in the last round of computing, and p_{i-2} is the number of pending points in the penultimate computing, x is an accommodation coefficient as $(r_{i-1} - r_{i-2}) \times (p_{i-2} - p_{i-1})$, in which $r_{i-1} - r_{i-2}$ is used to obtain the change tendency of radius, and $p_{i-2} - p_{i-1}$ is used to quantify the effect of adjustment: $(r_{i-1} - r_{i-2}) > 0$ means that the last adjustment of r is increased, if $(p_{i-2} - p_{i-1}) > 0$ means pending points are

reduced and the tendency of adjustment should be maintained, otherwise if $(p_{i-2} - p_{i-1}) < 0$ means the adjustment should be turned; when $(r_{i-1} - r_{i-2}) < 0$, if pending points are reduced, $(p_{i-2} - p_{i-1}) > 0$ and then r will be reduced continually, otherwise if $(p_{i-2} - p_{i-1}) > 0$, the value of r will be increased to turn the tendency.

The points' distribution is shown in Figure 8 where the initial value of r is set as 0.25, and there are two pending points detected; then, by (12), the optimized value of r is 0.29, and the result of computing is shown in Figure 9, in which there are three pending points; then, the value of r is reduced to 0.21 by (12), and the result of computing is shown in Figure 10, in which there is no pending point, and the procedure of optimizing is complete, and all points are well clustered.

Although the result of Figure 10 is a particular case, DDPA-DP has a suspension method to avoid increasing the time complexity: if there are C continuous rounds of computing with same number of pending points or the changing range of pending points is less than $1/C$, the optimizing procedure will be completed, in which C is the number of centers. If there are still some pending points after optimizing r , they will be analyzed to be divided to borders or outliers by the next two principles: if a point is a pending point and its nearest neighbor with larger ρ is an outlier, this pending point is also an outlier; if a point is a pending point, and its nearest neighbor point with larger ρ is a center or border, this point can be seemed as a border point.

3.3. The Complexity of DDPA-DP. Time complexity is an important performance in designing clustering algorithm because there are a large number of data sets to be computed. In DDPA-DP, n points are used to accomplish initial local density computing by initial parameter, and the fitted curves are established, so the complexity of this step is $O(n^2)$; and then in the local field radius optimizing step, just pending points should be redetected and its average number assumes p , and the complexity of this step is $O(p^*k)$, where k is the average computing rounds' number; then, at the last step, the optimized r is used to compute final points' roles and clustering, and the complexity is $O(n^2)$ too. Because the numbers of p and k are much less than n , the complexity of whole DDPA-DP is $O(n^2)$, which is the same with DP-Clust. Based on this analysis, it can be concluded that DDPA-DP can maintain relative high performance in complexity with parameter independence.

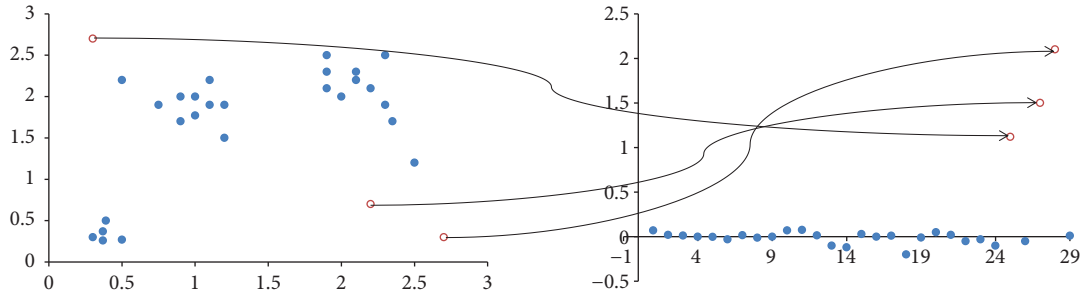


FIGURE 6: The corresponding points of outliers.

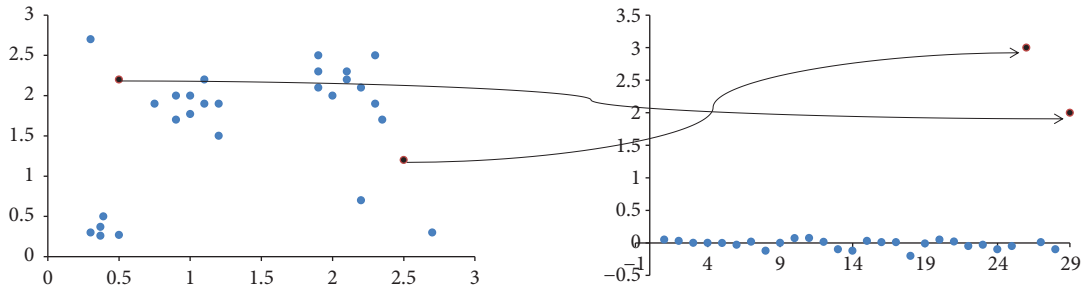


FIGURE 7: The corresponding points of pending points.

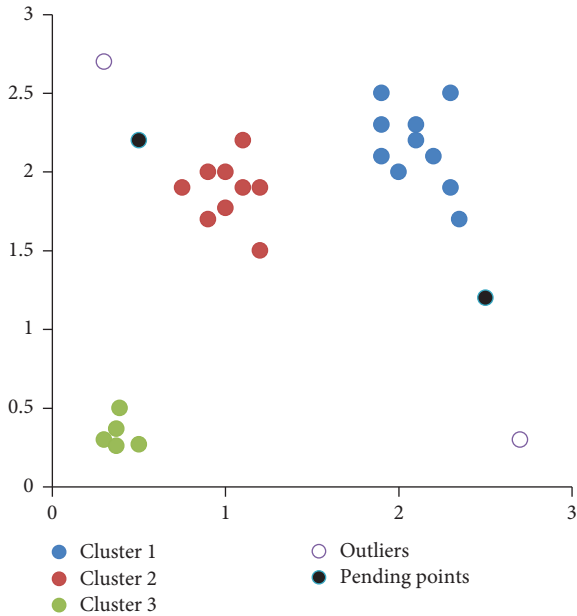


FIGURE 8: The distribution of points' roles when r is 0.25.

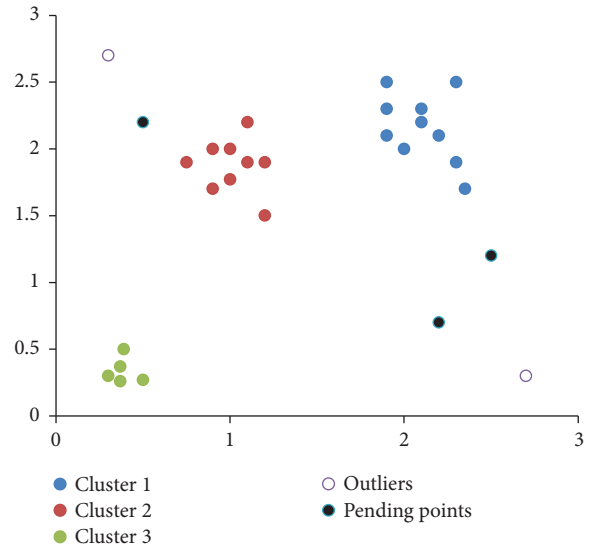


FIGURE 9: The distribution of points' roles when r is 0.29.

4. Experiments and Results

To prove the advantages of DDPA-DP, a series of experiments are designed and simulated, and three typical clustering algorithms DBSCAN [12], DP-Clust [21], DPC-KNN [26], FKNN-DPC [31], Dcore [32], and DCNaN [33] are compared with DDPA-DP in these experiments. In this section, experiments are simulated by MATLAB 2015b, and

two main performances are analyzed: the accuracies of all clustering algorithms are compared and analyzed in Section 4.1, and the real-time performances of these algorithms are compared and analyzed in Section 4.2. Six artificial data sets with arbitrary shapes of distribution and one real data set GL1 are used to be simulated, which are listed in Table 1. In Table 1, N means the number of data points, K means the number of clusters, and D means the number of dimensions. The distributions of 2-D data sets are shown in Figures 11–14.

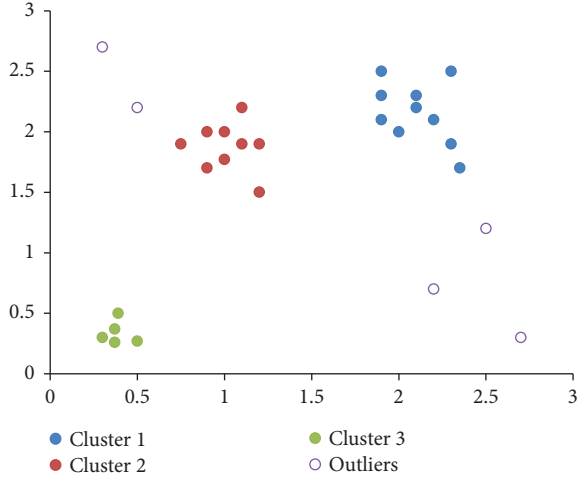
FIGURE 10: The distribution of points' roles when r is 0.21.

TABLE 1: The parameters of experiment data set.

Data set	N	K	D
Aggregation	788	7	2
Pathbased	300	3	2
Spiral	312	3	2
Jain	373	2	2
DIM	1024	16	32
KDDCUP04Bio	145751	2000	74
GL1	280307	16	18

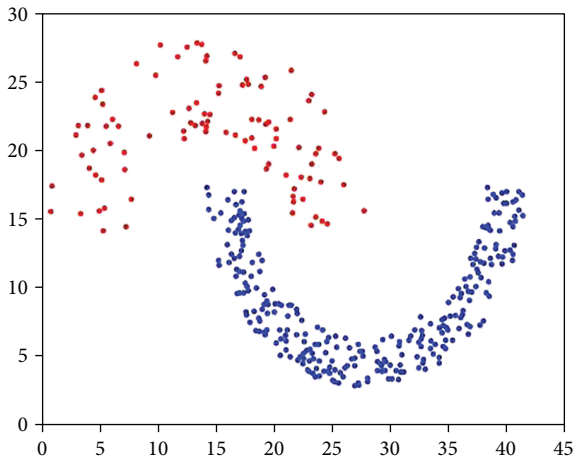


FIGURE 11: The distribution of JAIN.

4.1. *The Accuracy of Clustering.* Accuracy is one of the most important performances of clustering algorithms, and to compare different algorithms' clustering accuracy, the clustering purities of all algorithms in the same data set are calculated, and clustering purity has been used in

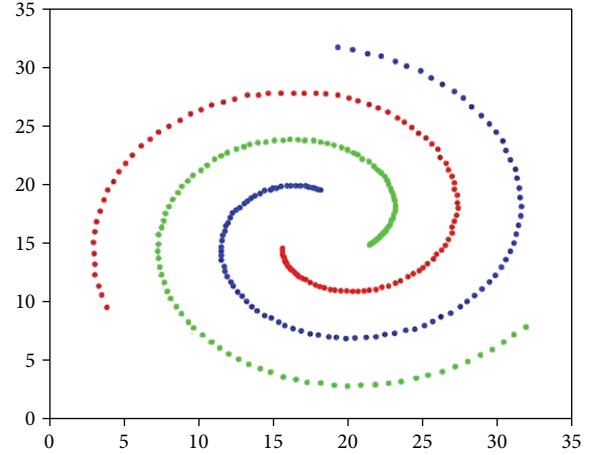


FIGURE 12: The distribution of Spiral.

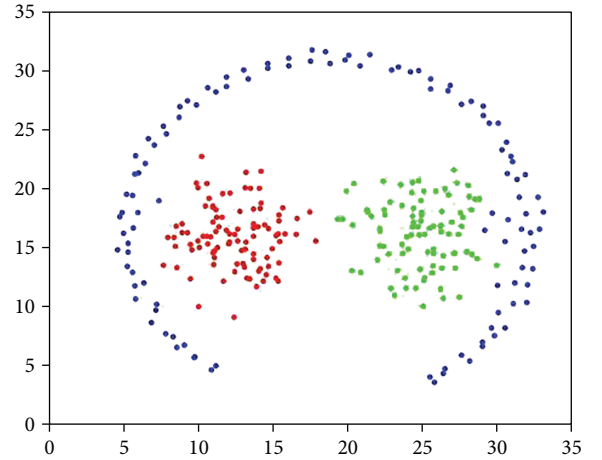


FIGURE 13: The distribution of Pathbased.

most researches to judge clustering accuracy [24, 26, 27, 32].
purity = $\sum_{i=1}^k (|C_i^d|/|C_i|/K)$ in (13).

$$\text{purity} = \sum_{i=1}^k \frac{|C_i^d|/|C_i|}{K}. \quad (13)$$

In (13), K is the number of clusters, and $|C_i^d|$ represents the number of data points correctly distributed to cluster i , and $|C_i|$ represents the number of all data points in cluster i . Then, purity is the ratio of correctly clustered to all points, and its value is between 0 and 1. The ratio is higher, the clustering result is more accurate, so it can be used to directly illustrate the different algorithms' performance in the same data set, and the experiments in this section are compared based on clustering purity.

Before clustering, initial parameters should be preset to deal with different data sets, and in DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP, the initial parameter should be set is the radius of local field, and in DPC-KNN

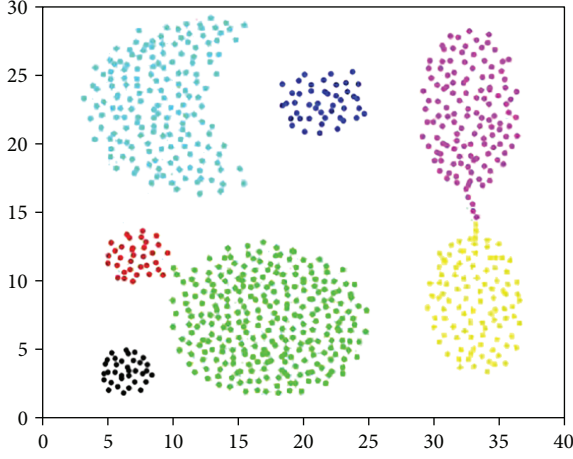


FIGURE 14: The distribution of Aggregation.

and FKNN-DPC, the initial parameter should be set is the neighbors' number of every point. In this paper, the initial parameters are defined also by data driven instead of by experience, and to overall simulate different applications, three initial states are designed in this section to better illustrate the parameter independence and accuracy of different algorithms. The first parameter state is set that the local field's radius $r1$ is as (14), and the neighbors' number $K1$ is by as (15).

$$r1 = \sum_{x=1}^C \sum_{i=1, j=1, i \neq j}^{n_x} \frac{2d_{ij}}{n_x C}, \quad (14)$$

$$K1 = \sum_{j=1}^C \sum_{i=1}^N \frac{x_i}{C}, \quad x_i = \begin{cases} 1, & d(i, c_j) < r1, \\ 0, & d(i, c_j) > r1. \end{cases} \quad (15)$$

In (14), C is the number of clusters in the data set, and n_x is the number of points in x th cluster, and it can be seen that $r1$ is computed by the average distance between points in the same clusters. Meanwhile in (15), N is the number of all points, and c_j is the center of cluster j , and if point i is located in the local field of c_j , x_i is set 1, otherwise it is set 0. It can be seen that $K1$ is computed by the average number of points in the centers' local fields.

The second parameter state is as the local field's radius $r2$ is as (16), in which the parameters' means are the same as (14), and it can be seen that $r2$ is computed by the average distance among all points in the data set. The neighbors' number $K2$ is as (17), and it can be seen that $K2$ is decided by the average number of points in all points' local fields.

$$r2 = \sum_{i=1, j=1, i \neq j}^N \frac{2d_{ij}}{N}, \quad (16)$$

$$K2 = \sum_{j=1}^N \sum_{i=1, i \neq j}^N \frac{x_i}{N}, \quad x_i = \begin{cases} 1, & d(i, j) < r2, \\ 0, & d(i, j) > r2. \end{cases} \quad (17)$$

The third parameter state is as the local field's radius $r3$ is decided by the average distance among the points in different clusters as (18), and the neighbors' number K is decided by the average number of points in all points' local fields too.

$$r3 = \sum_{C_x=1}^C \sum_{C_y=C_x+1}^C \sum_{i \in c_x, j \in c_y} \frac{2d_{ij}}{mn}. \quad (18)$$

In (18), m is the points' number in cluster c_x and n is the e points' number of c_y . It can be seen that $r3$ is computed by the average distance among the points in different clusters.

Among three initial local field's radiuses, $r1$ is the least because the distances between points in the same cluster are obviously less than the distances between different clusters as $r3$, and $r2$ is at the middle of $r1$ and $r3$, so by these three initial radiuses, DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP can be relatively overall simulated and compared. Meanwhile, the initial values of K are also divided to three levels: $K1$ is the largest one because it is decided by the neighbors of centers, and centers have obviously more neighbors than other points; then, $K2$ is decided by the average neighbors of all points, and borders and outliers have less neighbors than centers, so it is obviously less than $K1$; $K3$ is at the middle of $K1$ and $K2$, because when computing $K3$, the points' local field is expanded, so it is larger than $K2$, however, it is also decided by all points' neighbors, and then it is less than $K1$. By these three levels of K , DPC-KNN and FKNN-DPC are also able to be overall compared.

In Figure 15, the clustering results of these algorithms for JAIN are shown. JAIN 1 means the state the local field's radius r in DBSCAN, DP-Clust, DCore, DCNaN, and DDPA-DP is set as $r1$ with value 2 computed by (14), and the neighbors' number K in DPC-KNN and FKNN-DPC is set as $K1$ with value 70 by (15); JAIN 2 means the state the local field's radius is set as set as $r2$ with value 2.5 by (16), and the neighbors' number is $K2$ with value 50 by (17); JAIN 3 means the state the local field's radius is set as $r3$ with value 2.75 by (18), and the neighbors' number is set as $K3$ with value 60 by (17) with $r2$. By Figure 15, it can be concluded that DDPA-DP has obvious advantage in the accuracy of clustering no matter what initial states set, and its accuracy is not less than 0.96; DCore and DCNaN have relative stable accuracy, but they have no advantage over DPC-KNN and FKNN-DPC; DPC-KNN and FKNN-DPC are influenced by the value of K , and the larger accuracy will be obtained with larger K ; DP-Clust is much influenced by the value of r and it is just advanced than DBSCAN.

In Figure 16, the clustering accuracy of these algorithms in Spiral is shown, and because the distribution of Spiral is much unbalance and there are some sparse areas, the clustering accuracies of most algorithms are declined; however, DDPA-DP can still maintain the accuracy is not less than 0.95, and it is not much influenced by initial set of r . The results of DP-Clust, DPC-KNN, and FKNN-DPC are influenced by initial parameter much obviously, and the results of DCore and DCNaN are also obviously influenced by

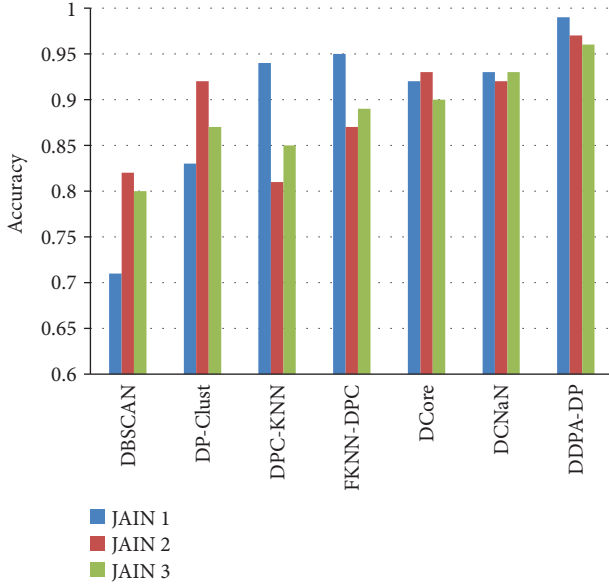


FIGURE 15: The simulation results in JAIN.

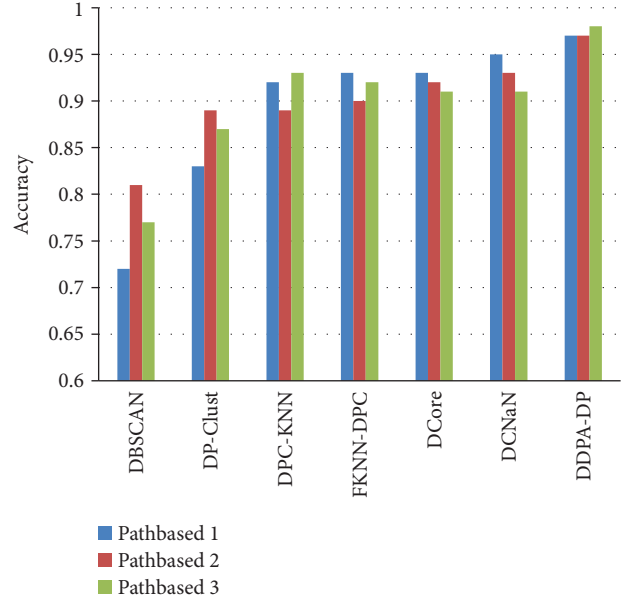


FIGURE 17: The simulation results in Pathbased.

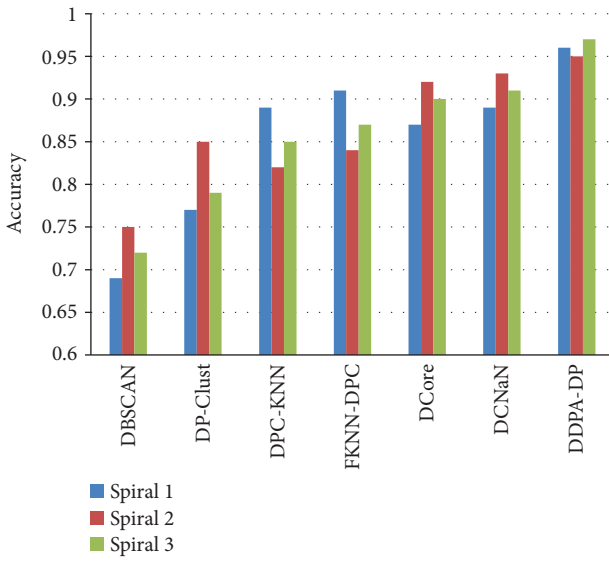


FIGURE 16: The simulation results in Spiral.

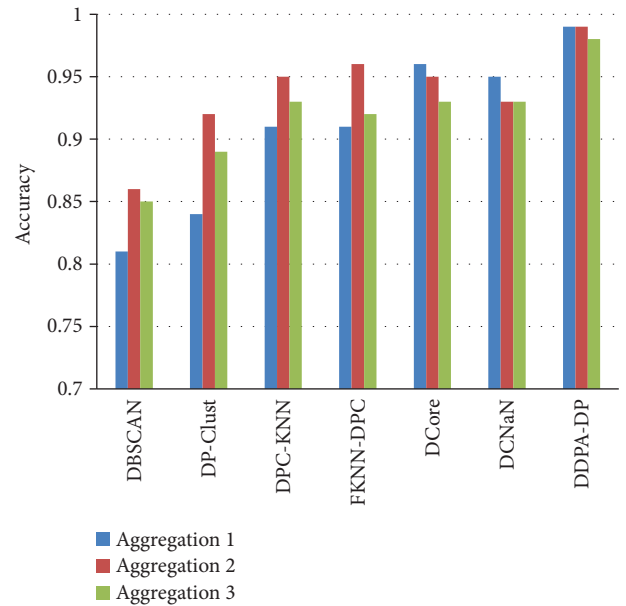


FIGURE 18: The simulation results in Aggregation.

parameter. The initial parameters in Spiral 1 is that the local field's radius r_1 is set as 1.5, and the neighbors' number K_1 is set as 75; Spiral 2 means the state r_2 is set as 1.7 and K_2 is set as 50; Spiral 3 means the state r_3 is set as 2 and K_3 is set as 65.

In Figures 17 and 18, the clustering accuracies in Pathbased data set and Aggregation data set of these algorithms are shown. Although the density of these two data sets are relatively uniform, but the shapes of clusters are arbitrary. It can be seen that the accuracies of DDPA-DP are both stable and in a relative high level; and other algorithms' performances are not stable especially in Aggregation. According to (14), (15), (16), (17), and (18), in the initial parameters, Pathbased 1 means the local field's radius r_1 is set as 5, and the neighbors' number K_1 is set as 60; Pathbased 2 means r_2 is set as 3, and the neighbors' number K_2 is set as

50; Pathbased 3 means r_3 is set as 4, and the neighbors' number K_3 is set as 55. Then, the initial parameters in Aggregation 1 are that the local field's radius r is set as 2, and the neighbors' number K is set as 100; in Aggregation 2, the local field's radius r is set as 2.5 and the neighbors' number K is set as 85; in Aggregation 3, the local field's radius r is set as 3 and the neighbors' number K is set as 75.

In Figures 19–21, the clustering results of these algorithms in high-dimension data sets DIM, KDDCUP04Bio, and GL1 are shown, and in these data sets, the accuracies of all algorithms are declined obviously. In DIM, the accuracies of DBSCAN and DP-Clust are less than 0.75 in all

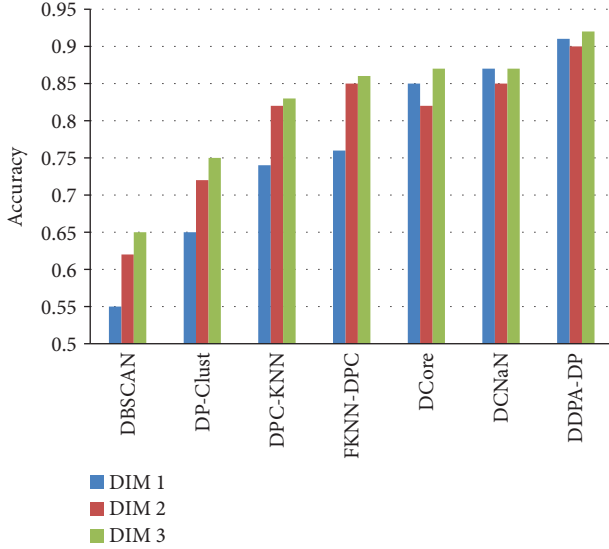


FIGURE 19: The simulation results in DIM.

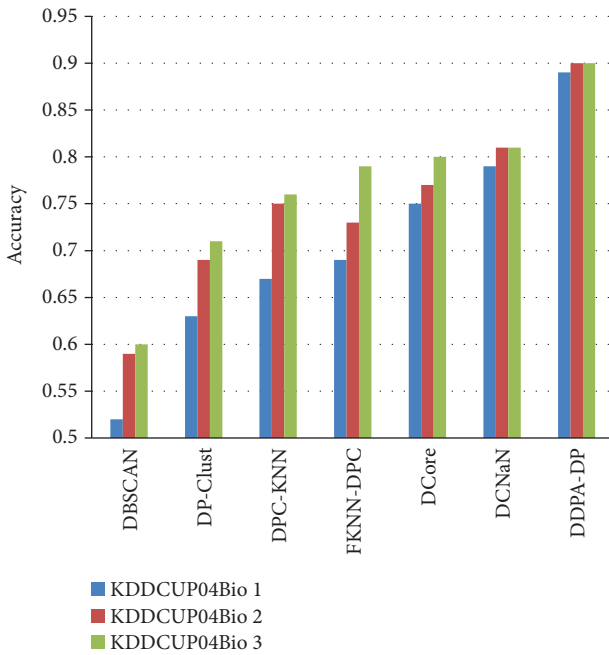


FIGURE 20: The simulation results in KDDCUP04Bio.

states; and although DPC-KNN and FKNN-DPC have relative large accuracies, but they are much influenced by the value of K , and the range abilities are larger than 10%; although DCore and DCNaN have stable accuracies, they are still less than DDPA-DP, and just DDPA-DP can obtain the clustering accuracy larger than 0.9. In DIM 1, the local field's radius r is set as a 32-dimension vector with value 5 and K is set as 100; in DIM 2, the local field's radius r is set as 6, and K is set as 120; in DIM 3, r is set as 7, and K is set as 150.

In Figure 20, the accuracies of clustering are further down because of the large number of data points in KDDCUP04Bio

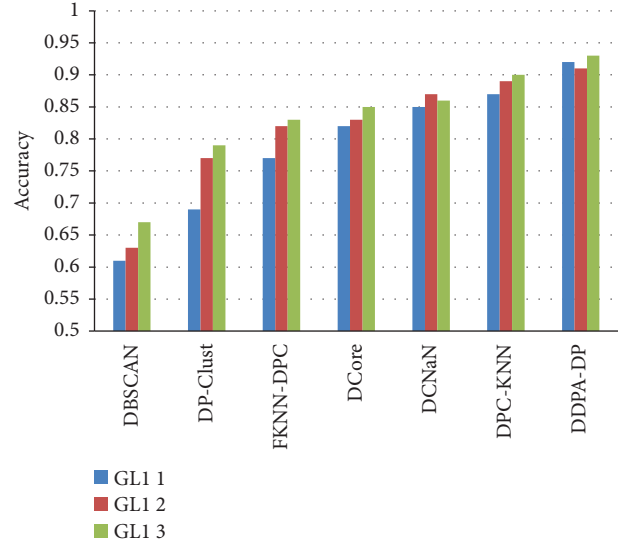


FIGURE 21: The simulation results in GL 1.

with high dimensions. By the results of Figure 20, the advantage of DDPA-DP is more obvious with the increase of points, and there are no any algorithms that can obtain the clustering accuracies larger than 0.8 except DDPA-DP, and DDPA-DP can maintain the accuracy as 0.9 in every state. In this data set, there are too many points to obtain the exact parameters by the method as the used in last five data sets, so we define the parameters by random sampling 20% points from all. Then by (14) to (18), in KDDCUP04Bio 1, the local field's radius r_1 is set as a 74-dimension vector with value 20, and K_1 is set as 2000; in KDDCUP04Bio 2, r_2 is set as 27, and K_2 is set as 1500; in KDDCUP04Bio 3, r_3 is set as 35, and K_3 is set as 1200.

In Figure 21, the data set GL1 is a real data collected from a thermal power plant, and the distribution of GL1 is more uniform than KDDCUP04Bio although it has more points. By Figure 21, all algorithms' accuracies are improved than KDDCUP04Bio, and the advantage of DDPA-DP is not obvious as KDDCUP04Bio, but it is still the most acute and stable one. The parameters in GL1 are set by the same method as in KDDCUP04Bio. In GL1 1, r_1 is set as an 18-dimension vector with value 17, and K_1 is set as 1750; in GL1 2, r_2 is set as 21, and K_2 is set as 1395; In GL1 3, r_3 is set as 27, and K_3 is set as 1535.

Based on the simulated results in this section, it can be concluded that DDPA-DP has obvious advantage in clustering accuracy, because the parameters in DDPA-DP are continuously adapted by the data-driven method, by which the parameters are optimized to improve the clustering accuracy, and then the optimized parameters can reduce the influence by initial set values which ensures the clustering accuracy is stable at high level. The advantages of DDPA-DP are more obvious with more complex data set, so DDPA-DP is fitter for the big data applications.

4.2. The Runtime of Clustering. Runtime is also an important standard to estimate the performance of clustering algorithm, and it can be used to estimate the time complexity of

clustering. In this section, the runtime of DDPA-DP is compared with DBSCAN, DP-Clust, DPC-KNN, FKNN-DPC, DCore, and DCNaN, and the results of experiments are shown in Figures 22 and 23. These results are obtained by computing the average runtime of every algorithm in the three states introduced in Section 4.1.

In Figure 22, the results of these algorithms simulated in relative small data sets are shown. From these results, it can be concluded that DP-Clust has the best time complexity of all, because the calculation procedure of DP-Clust is the simplest; among other algorithms, the runtimes have little differences, because in small data sets, the calculation procedure of iterating, optimizing, and searching neighbor points in local field can be accomplished in a short time.

In Figure 23, the runtimes of these algorithms in two large data sets are shown. Based on these results, it can be seen that the runtime of DDPA-DP has become obvious less than other ones except DBSCAN and DP-Clust. In DBSCAN and DP-Clust, all clustering operations are executed one time, which reduces their time complexity; in DCore and DCNaN, large data set means there will be many “false peaks” when detecting points’ roles, and the discovery of density core needs many comparing operations, so much iteration will be executed in these two algorithms, which leads the runtime of DCore and DCNaN are the longest ones; in DPC-KNN and FKNN-DPC, the K neighbors should be used to judge the local fields for every point, and to obtain high clustering accuracy, the value of K is generally large in large data sets, and meanwhile, the iteration should be executed to optimize the choose of neighbors, which leads the time complexities of these two algorithms are just less than DCore and DCNaN and larger than others; in DDPA-DP, the optimization of local field radius r is determined by the distribution of “pending points,” and these points are small in number among all points especially in large data sets, and the calculation of detecting pending points is much less than detecting other roles, so the time complexity in iteration is not much increased, and its runtime will be close to DP-Clust with high clustering accuracy.

5. Conclusion

Based on the classical density-based clustering algorithm DP-Clust, we proposed a parameter adaptive clustering algorithm named DDPA-DP in this paper. The data-driven thought goes through the design of DDPA-DP: at first, a series of fitted curves are established to automatically detect points’ roles by points’ density attributes instead of any artificial thresholds; meanwhile, a new point’s role “pending point” is defined, and then by the change of pending points’ number, the local field’s radius can be adaptively optimized.

DDPA-DP improves the flexibility of clustering by avoiding the influence of artificial parameters, and the time complexity of DDPA-DP is not significantly increased comparing with DP-Clust because there is little extra calculation added to optimize parameters. A series of experiments are designed to compare DDPA-DP with some existing clustering algorithms, and in these experiments, some typical synthetic data sets and a real-world data set from thermal

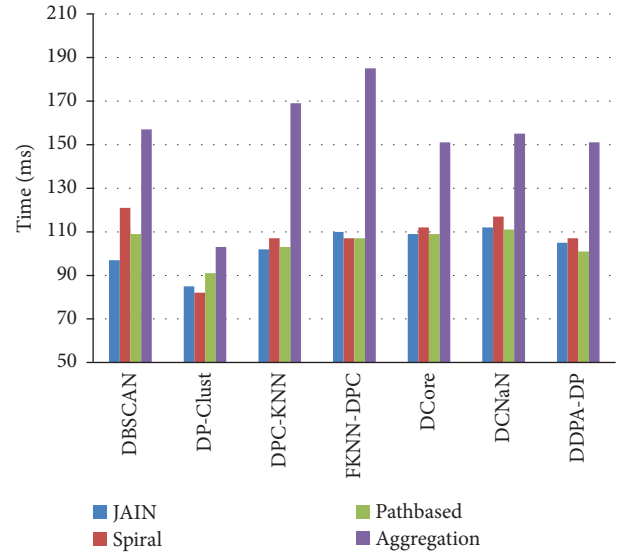


FIGURE 22: The runtimes in small data sets.

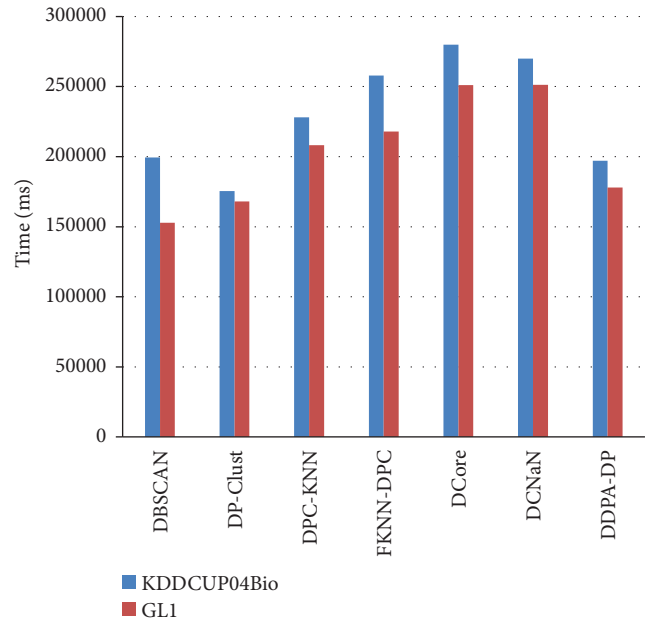


FIGURE 23: The runtimes in large data sets.

power industry are simulated with different initial conditions to overall estimate these algorithms. By the results of experiments, it can be concluded that DDPA-DP has advantage in the performance of clustering accuracy and time complexity.

Data Availability

All artificial data sets can be downloaded from the following website: <http://cs.uef.fi/sipu/datasets/>.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by Natural Science Foundation of China under Contract no. 60573065 and Science and Technology Development Plan of Shandong Province under Contract no. 2014GGX101039, and it is partially supported by Natural Science Foundation of China under Contract no. 60903176.

References

- [1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [2] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, no. 2-3, pp. 293–306, 1985.
- [3] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.
- [4] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.
- [5] A. Cornuéjols, C. Wemmert, P. Gançarski, and Y. Bennani, "Collaborative clustering: why, when, what and how," *Information Fusion*, vol. 39, pp. 81–95, 2018.
- [6] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.
- [7] J. B. Mac Queen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, 1967.
- [8] T. Velmurugan and T. Santhanam, "Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points," *Journal of Computer Science*, vol. 6, no. 3, pp. 363–368, 2010.
- [9] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *ACM SIGMOD Record*, vol. 25, no. 2, pp. 103–114, 1996.
- [10] S. Guha, R. Rastogi, and K. Shim, "ROCK: a robust clustering algorithm for categorical attributes," *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, 1999, pp. 512–521, Sydney, NSW, Australia, 1999.
- [11] G. Karypis, Eui-Hong Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
- [12] M. Ester, H. P. Kriegel, J. Sander, and X. W. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, Portland, OR, USA, 1996.
- [13] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *SIGMOD '99 Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, vol. 28, no. 2, pp. 49–60, Philadelphia, Pennsylvania, USA, May-June 1999.
- [14] W. Wang, J. Yang, and R. R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining," in *Proceedings of International Conference on Very Large Data Bases*, pp. 186–195, Athens, Greece, 1997.
- [15] M. Ankerst, M. Breunig, and H. Kriegel, "OPTICS: Ordering points to identify the clustering structure," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 49–60, ACM Press, New York, USA, 1999.
- [16] C. M. M. Pereira and R. F. de Mello, "PTS: Projected Topological Stream clustering algorithm," *Neurocomputing*, vol. 180, pp. 16–26, 2016.
- [17] E. W. M. Ma and T. W. S. Chow, "A new shifting grid clustering algorithm," *Pattern Recognition*, vol. 37, no. 3, pp. 503–514, 2004.
- [18] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 133–142, San Jose, California, USA, 2007.
- [19] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 3, pp. 1–28, 2009.
- [20] R. M. Aliguliyev, "Performance evaluation of density-based clustering methods," *Information Sciences*, vol. 179, no. 20, pp. 3583–3602, 2009.
- [21] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
- [22] J. de Andrade Silva, E. R. Hruschka, and J. Gama, "An evolutionary algorithm for clustering data streams with a variable number of clusters," *Expert Systems with Applications*, vol. 67, pp. 228–238, 2017.
- [23] J. Hou and W. Liu, "Parameter independent clustering based on dominant sets and cluster merging," *Information Sciences*, vol. 405, pp. 1–17, 2017.
- [24] J. Nordhaug Myhre, K. Øyvind Mikalsen, S. Løkse, and R. Jenssen, "Robust clustering using a kNN mode seeking ensemble," *Pattern Recognition*, vol. 76, pp. 491–505, 2018.
- [25] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.
- [26] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.
- [27] C. Jinyin, L. Xiang, Z. Haibing, and B. Xintong, "A novel cluster center fast determination clustering algorithm," *Applied Soft Computing*, vol. 57, pp. 539–555, 2017.
- [28] J.-Y. Chen and H.-H. He, "A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data," *Information Sciences*, vol. 345, pp. 271–293, 2016.
- [29] F. Saki and N. Kehtarnavaz, "Online frame-based clustering with unknown number of clusters," *Pattern Recognition*, vol. 57, pp. 70–83, 2016.
- [30] J. Xu, G. Wang, T. Li, W. Deng, and G. Gou, "Fat node leading tree for data stream clustering with density peaks," *Knowledge-Based Systems*, vol. 120, pp. 99–117, 2017.
- [31] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

- [32] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.
- [33] Y. Chen, S. Tang, L. Zhou et al., "Decentralized clustering by finding loose and distributed density cores," *Information Sciences*, vol. 433-434, pp. 510–526, 2018.
- [34] J. Xie, Z.-Y. Xiong, Y.-F. Zhang, Y. Feng, and J. Ma, "Density core-based clustering algorithm with dynamic scanning radius," *Knowledge-Based Systems*, vol. 142, pp. 58–70, 2018.