CrossMark

JM PAPERS

Sparse identification of nonlinear dynamics with low-dimensionalized flow representations

Kai Fukami[1,2,†], **Takaaki Murata**[2], **Kai Zhang**[3] **and Koji Fukagata**[2]

[1]Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA

[2]Department of Mechanical Engineering, Keio University, Yokohama 223-8522, Japan

[3]Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854, USA

We perform a sparse identification of nonlinear dynamics (SINDy) for low-dimensionalized complex flow phenomena. We first apply the SINDy with two regression methods, the thresholded least square algorithm and the adaptive least absolute shrinkage and selection operator which show reasonable ability with a wide range of sparsity constant in our preliminary tests, to a two-dimensional single cylinder wake at $Re_D = 100$, its transient process and a wake of two-parallel cylinders, as examples of high-dimensional fluid data. To handle these high-dimensional data with SINDy whose library matrix is suitable for low-dimensional variable combinations, a convolutional neural network-based autoencoder (CNN-AE) is utilized. The CNN-AE is employed to map a high-dimensional dynamics into a low-dimensional latent space. The SINDy then seeks a governing equation of the mapped low-dimensional latent vector. Temporal evolution of high-dimensional dynamics can be provided by combining the predicted latent vector by SINDy with the CNN decoder which can remap the low-dimensional latent vector to the original dimension. The SINDy can provide a stable solution as the governing equation of the latent dynamics and the CNN-SINDy-based modelling can reproduce high-dimensional flow fields successfully, although more terms are required to represent the transient flow and the two-parallel cylinder wake than the periodic shedding. A nine-equation turbulent shear flow model is finally considered to examine the applicability of SINDy to turbulence, although without using CNN-AE. The present results suggest that the proposed scheme with an appropriate parameter choice enables us to analyse high-dimensional nonlinear dynamics with interpretable low-dimensional manifolds.

Key words: computational methods, machine learning, low-dimensional models

† Email address for correspondence: kfukami1@g.ucla.edu

1. Introduction

Sparse identification of nonlinear dynamics (SINDy) (Brunton, Proctor & Kutz 2016*a*) is one of the prominent data-driven tools to obtain governing equations of nonlinear dynamics in a form that we can understand. The SINDy algorithm enables us to discover a governing equation from time-discretized data and identify dominant terms from a large set of potential terms that are likely to be involved in the model. Recently, the usefulness of the SINDy has been demonstrated in various fields (Champion, Brunton & Kutz 2019*b*; Hoffmann, Fröhner & Noé 2019; Zhang & Schaeffer 2019; Deng *et al.* 2020). Here, let us introduce some efforts, especially in the fluid dynamics community. Loiseau, Noack & Brunton (2018*b*) utilized SINDy to present general reduced-order modelling (ROM) framework for experimental data: sensor data and particle image velocimetry data. The model was investigated using a transient and post-transient laminar cylinder wake. They reported that the nonlinear full-state dynamics can be modelled with sensor-based dynamics and SINDy-based estimation for coefficients of ROM. The SINDy which takes into account control inputs (called SINDYc) was also investigated by Brunton, Proctor & Kutz (2016*b*) using the Lorenz equations. Loiseau & Brunton (2018) combined SINDy and proper orthogonal decomposition (POD) to enforce energy-preserving physical constraints in the regression procedure toward the development of a new data-driven Galerkin regression framework. For the time-varying aerodynamics, Li *et al.* (2019) identified vortex-induced vibrations on a long-span suspension bridge utilizing the SINDy algorithm extended to parametric partial differential equations (Rudy *et al.* 2017). As a novel method to perform the order reduction of data and SINDy simultaneously, there is a customized autoencoder (AE) (Champion *et al.* 2019*a*) introducing the SINDy loss in the loss function of deep AE networks. The sparse regression idea has also recently been propagated to turbulence closure modelling purposes (Beetham & Capecelatro 2020; Schmelzer, Dwight & Cinnella 2020; Beetham, Fox & Capecelatro 2021; Duraisamy 2021). As reported in these studies, by employing the SINDy to predict the temporal evolution of a system, we can obtain ordinary differential equations, which should be helpful to many applications, e.g. control of a system. In this way, the propagation of the use of SINDy can be seen in the fluid dynamics community.

We here examine the possibility of SINDy-based modelling of low-dimensionalized complex fluid flows presented in figure 1. Following our preliminary tests with the van der Pol oscillator and the Lorenz attractor presented in the Appendices, we apply the SINDys with two regression methods, (1) thresholded least square algorithm (TLSA) and (2) adaptive least absolute shrinkage and selection operator (Alasso), to a two-dimensional cylinder wake at $Re_D = 100$, its transient process, and a wake of two-parallel cylinders as examples of high-dimensional dynamics. In this study, a convolutional neural network-based autoencoder (CNN-AE) is utilized to handle these high dimensional data with SINDy whose library matrix is suitable for low-dimensional variable combinations (Kaiser, Kutz & Brunton 2018). The CNN-AE here is employed to map a high-dimensional dynamics into low-dimensional latent space. The SINDy is then utilized to obtain a governing equation of the mapped low-dimensional latent vector. Unifying the dynamics of the latent vector obtained via SINDy with the CNN decoder which can remap the low-dimensional latent vector to the original dimension, we can present the temporal evolution of high-dimensional dynamics and avoid the issue with high-dimensional variables as discussed above. The scheme of the present ROM is inspired by Hasegawa *et al.* (2019, 2020*a*,*b*) who utilized a long short term memory instead of SINDy in the present ROM framework. The paper is organized as follows. We provide details of SINDy and CNN-AE in § 2. We present the results for high-dimensional flow examples with the CNN-AE/SINDy ROM in §§ 3.1, 3.2 and 3.3. In § 3.4, we also discuss
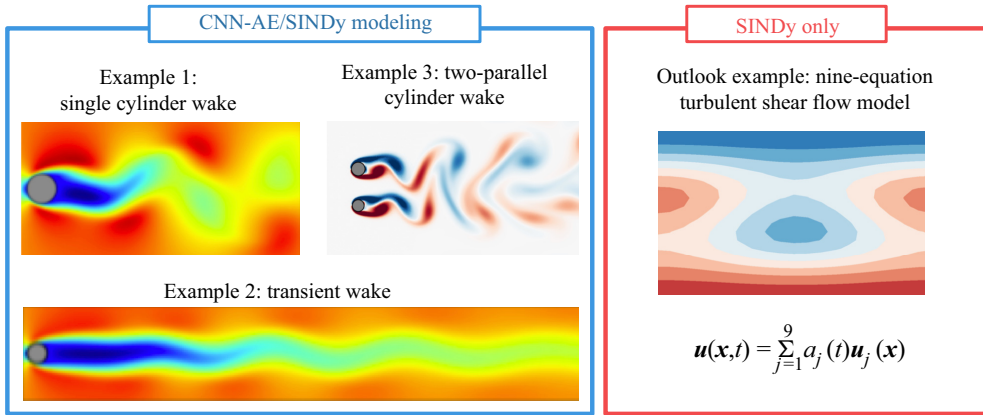
Figure 1. Covered examples of fluid flows in the present study.

its applicability to turbulence using the nine-equation turbulent shear flow model, although without considering the CNN-AE. At last, concluding remarks with outlook are offered in § 4.

2. Methods

2.1. *SINDy*

Sparse identification of nonlinear dynamics (Brunton *et al.* 2016*a*) is performed to identify nonlinear governing equations from time series data in the present study. The temporal evolution of the state $x(t)$ in a typical dynamical system can often be represented in the form of an ordinary differential equation,

$$\dot{x}(t) = f(x(t)). \tag{2.1}$$

To explain the SINDy algorithm, let $x(t)$ be $(x(t), y(t))$ hereinafter, although the SINDy can also handle the dynamics with higher dimensions, as will be considered later. First, the temporally discretized data of x are collected to arrange a data matrix X,

$$X = \begin{pmatrix} x^T(t_1) \\ x^T(t_2) \\ \vdots \\ x^T(t_m) \end{pmatrix} = \begin{pmatrix} x(t_1) & y(t_1) \\ x(t_2) & y(t_2) \\ \vdots & \vdots \\ x(t_m) & y(t_m) \end{pmatrix}. \tag{2.2}$$

We then collect the time series data of the time-differentiated value $\dot{x}(t)$ to construct a time-differentiated data matrix $\dot{X}$,

$$\dot{X} = \begin{pmatrix} \dot{x}^T(t_1) \\ \dot{x}^T(t_2) \\ \vdots \\ \dot{x}^T(t_m) \end{pmatrix} = \begin{pmatrix} \dot{x}(t_1) & \dot{y}(t_1) \\ \dot{x}(t_2) & \dot{y}(t_2) \\ \vdots & \vdots \\ \dot{x}(t_m) & \dot{y}(t_m) \end{pmatrix}. \tag{2.3}$$

In the present study, the time-differentiated values are obtained with the second-order central-difference scheme. Note in passing that results in the present study are not sensitive

to the differential scheme with appropriate time steps for construction of $\dot{X}$. Then, we prepare a library matrix $\Theta(X)$ including nonlinear terms of X,

$$\Theta(X) = \begin{pmatrix} | & | & | & | & | & | \\ 1 & X & X^{P_2} & X^{P_3} & X^{P_4} & X^{P_5} \\ | & | & | & | & | & | \end{pmatrix}, \tag{2.4}$$

where X^{P_i} is ith-order polynomials constructed by x and y. The set of nonlinear potential terms here includes up to fifth-order terms, although what terms are included can be arbitrarily set. We finally determine a coefficient matrix Ξ,

$$\Xi = (\xi_x \quad \xi_y) = \begin{pmatrix} \xi_{(x, 1)} & \xi_{(y, 1)} \\ \xi_{(x, 2)} & \xi_{(y, 2)} \\ \vdots & \vdots \\ \xi_{(x, l)} & \xi_{(y, l)} \end{pmatrix}, \tag{2.5}$$

in the state equation,

$$\dot{X}(t) = \Theta(X)\Xi, \tag{2.6}$$

using an arbitrary regression method, such as TLSA, least absolute shrinkage and selection operator (Lasso) and so on. The subscript l in (2.5) denotes the row index of the library matrix. Once the coefficient matrix Ξ is obtained, the governing equation is identified as

$$\dot{x} = \Theta(x)\xi_x, \quad \dot{y} = \Theta(y)\xi_y. \tag{2.7a,b}$$

In this study, we use the TLSA used in Brunton *et al.* (2016*a*) and the Alasso (Zou 2006) to obtain the coefficient matrix Ξ following our preliminary tests. Details can be found in the Appendices.

2.2. *CNN-AE*

We use a CNN (LeCun *et al.* 1998) AE (Hinton & Salakhutdinov 2006) for order reduction of high-dimensional data, as shown in figure 2. As an example, we show the CNN-AE with three hidden layers. In a training process, the AE $\mathcal{F}$ is trained to output the same data as the input data q such that $q \approx \mathcal{F}(q; w)$, where w denotes the weights of the machine learning model. The process to optimize the weights w can be formulated as an iterative minimization of an error function E,

$$w = \mathrm{argmin}_w[E(q, \mathcal{F}(q; w))]. \tag{2.8}$$

For the use of the AE as a dimension compressor, the dimension of an intermediate space called the latent space η is smaller than that of the input or output data q as illustrated in figure 2. When we are able to obtain the output $\mathcal{F}(q)$ similar to the input q such that $q \approx \mathcal{F}(q)$, it can be guaranteed that the latent vector r is a low-dimensional representation of its input or output q. In an AE, the dimension compressor is called the encoder $\mathcal{F}_e$ (the left-hand part in figure 2) and the counterpart is referred to as the decoder $\mathcal{F}_d$ (the right-hand part in figure 2). Using them, the internal procedure of the AE can be expressed as

$$r = \mathcal{F}_e(q), \quad q = \mathcal{F}_d(r). \tag{2.9a,b}$$

In the present study, the dimension of the latent space for the problems of a cylinder wake and its transient process is set to two following our previous work (Murata,
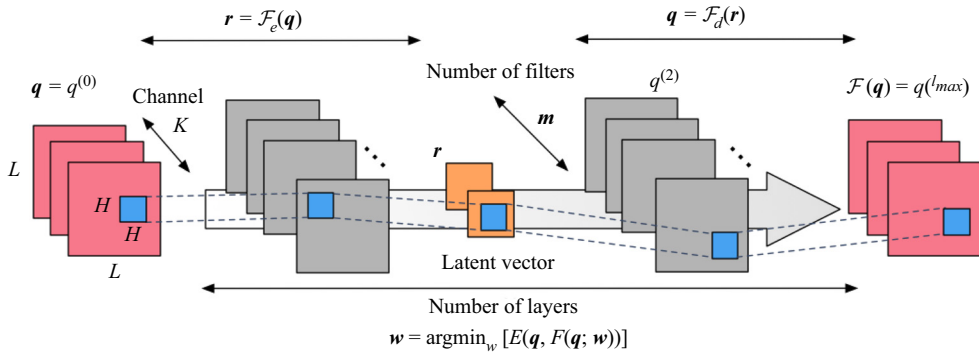
Figure 2. Convolutional neural network-based autoencoder.

Fukami & Fukagata 2020). Murata *et al.* (2020) reported that the flow fields can be mapped into a low-dimensional latent space successfully while keeping the information of high-dimensional flow fields. For the example of a two-parallel cylinder wake in § 3.3, the dimension of the latent space is set to four to handle its quasi-periodicity, accordingly. For construction of the AE, the L_2 norm error is applied as the error function E and the Adam optimizer (Kingma & Ba 2014) is utilized for updating the weights in the iterative training process.

Next, let us briefly explain the convolutional neural network. In the scheme of the convolutional neural network, a filter h, whose size is $H \times H$, is utilized as the weights w, as shown in figure 3(*a*). The mathematical expression here is

$$q_{ijm}^{(l)} = \psi \left(\sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{s=0}^{H-1} h_{pskm}^{(l)} q_{i+p-C,j+s-C,k}^{(l-1)} + b_{ijm}^{(l)} \right), \quad (2.10)$$

where $C = \text{floor}(H/2)$, $q^{(l)}$ is the output at layer l, and K is the number of variables in the data (e.g. $K = 1$ for black-and-white photos and $K = 3$ for RGB images). Although not shown in figure 3, b is a bias added to the results of the filter operation. The activation function ψ is usually a monotonically increasing nonlinear function. The AE can achieve more effective dimension reduction than the linear theory-based method, i.e. POD (Lumley 1967), thanks to the nonlinear activation function (Milano & Koumoutsakos 2002; Fukami *et al.* 2020*b*; Murata *et al.* 2020). In the present study, a hyperbolic tangent function $\psi(a) = (e^a - e^{-a}) \cdot (e^a + e^{-a})^{-1}$ is adopted for the activation function following Murata *et al.* (2020). Moreover, we use the convolutional neural network to construct the AE because the CNN is good at handling high-dimensional data with lower computational costs than fully connected models, i.e. multilayer perceptron, thanks to the filter concept which assumes that pixels of images have no strong relationship with those of far areas. Recently, the use of CNN has emerged to deal with high-dimensional problems including fluid dynamics (Fukami, Fukagata & Taira 2019*a,b*; Fukami *et al.* 2019*c*; Fukami, Fukagata & Taira 2020*a*; Fukami *et al.* 2021*b*; Omata & Shirayama 2019; Morimoto, Fukami & Fukagata 2020*a*; Fukami, Fukagata & Taira 2021*a*; Matsuo *et al.* 2021; Morimoto *et al.* 2021), although this concept was originally developed in the field of computer science.
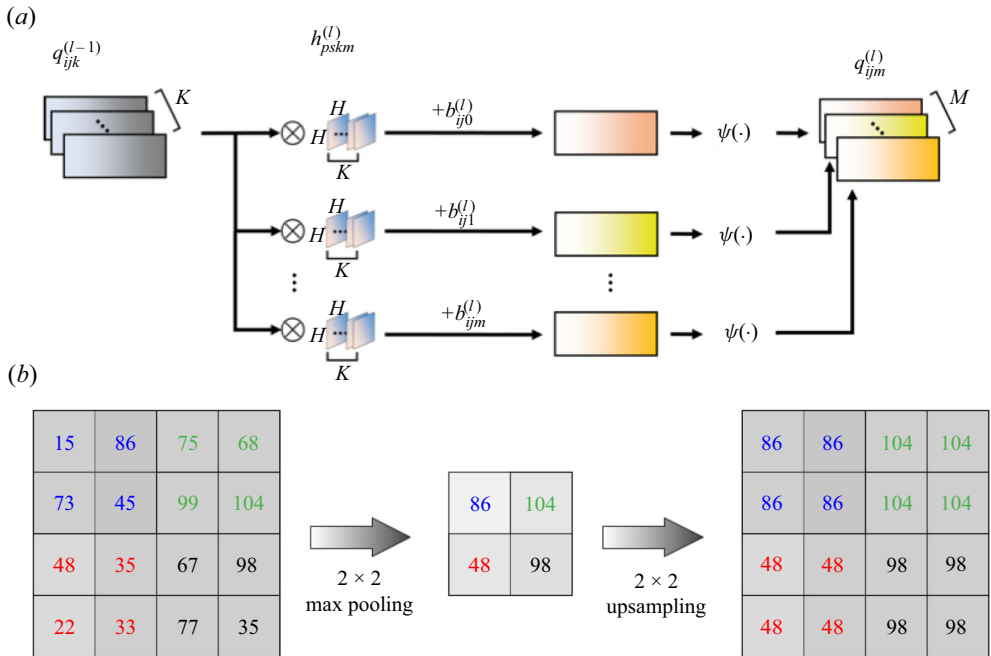
(*a*)

(*b*)

Figure 3. Internal procedure of convolutional neural network: (*a*) filter operation with activation function and (*b*) maximum pooling and upsampling.

2.3. *The CNN-SINDy-based ROM*

By combining the CNN-AE and SINDy, we present the machine-learning-based ROM (ML-ROM) as shown in figure 4. The CNN-AE first works to map a high-dimensional flow field into low-dimensional latent space. The SINDy is then performed to obtain a governing equation of the mapped low-dimensional latent vector. Unifying the predicted latent vector by the SINDy with the CNN decoder, we can obtain the temporal evolution of high-dimensional flow field as presented in figure 4. Moreover, the issue with high-dimensional variables and SINDy can also be avoided. Note again that the present ROM is inspired by Hasegawa *et al.* (2019, 2020*a*,*b*) who capitalized on a long short-term memory instead of SINDy. We apply this framework to a two-dimensional cylinder wake in § 3.1, its transient process in § 3.2 and a wake of two-parallel cylinders in § 3.3. In this study, we perform a fivefold cross-validation (Brunton & Kutz 2019) to create all machine learning models, although only the results of a single case will be presented, for brevity. The sample code for the present reduced-order model is available from https://github.com/kfukami/CNN-SINDy-MLROM.

3. Results and discussion

3.1. *Example 1: periodic cylinder wake*

Here, let us consider a two-dimensional cylinder wake using a two-dimensional direct numerical simulation (DNS). The governing equations are the incompressible continuity and Navier–Stokes equations,

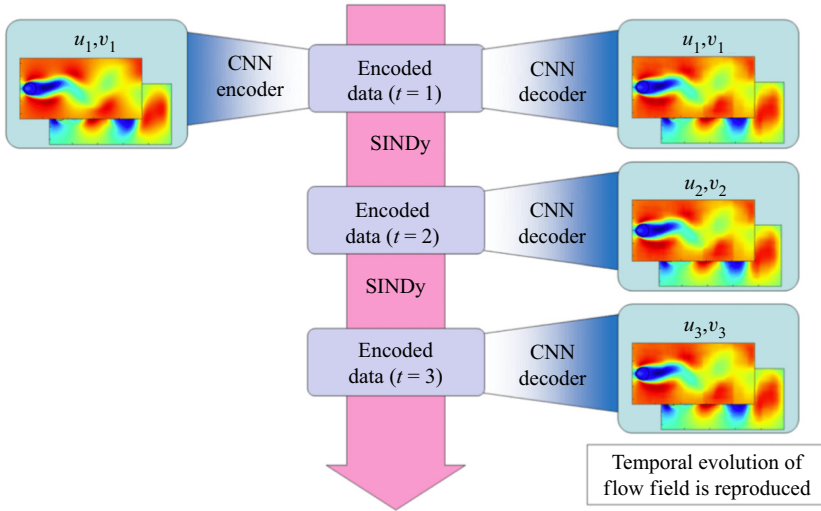$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.1}$$

Figure 4. The CNN-SINDy-based ROM for fluid flows.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{uu}) = -\boldsymbol{\nabla}p + \frac{1}{Re_D}\nabla^2\boldsymbol{u}, \tag{3.2}$$

where $\boldsymbol{u}$ and p represent the velocity vector and pressure, respectively. All quantities are made dimensionless by the fluid density, the free stream velocity and the cylinder diameter. The Reynolds number based on the cylinder diameter is $Re_D = 100$. The size of the computational domain is $L_x = 25.6$ and $L_y = 20.0$ in the streamwise (x) and the transverse (y) directions, respectively. The origin of coordinates is defined at the centre of the inflow boundary. A Cartesian grid with the uniform grid spacing of $\Delta x = \Delta y = 0.025$ is used; thus, the number of grid points is $(N_x, N_y) = (1024, 800)$. We use the ghost cell method (Kor, Badri Ghomizad & Fukagata 2017) to impose the no-slip boundary condition on the surface of cylinder whose centre is located at $(x, y) = (9, 0)$. In the present study, we utilize the flows around the cylinder as the training data set, i.e. $8.2 \leq x \leq 17.8$ and $-2.4 \leq y \leq 2.4$ with $(N_x^*, N_y^*) = (384, 192)$. The fluctuation components of streamwise velocity u and transverse velocity v are considered as the input and output attributes for CNN-AE. The time interval of the flow field data is $\Delta t = 0.025$ corresponding to approximately 230 snapshots per a period with the Strouhal number $St = 0.172$.

As mentioned above, the SINDy is employed to identify the ordinary differential equations that govern the temporal evolution of the mapped vector obtained by the CNN encoder. The time history and the trajectory of the mapped latent vector $\boldsymbol{r} = (r_1, r_2)$ are shown in figure 5. For the assessment of the candidate models, we integrate the differential equations to reproduce the time history. Note in passing that the indices based on the L_2 error are not suitable since the slight difference between the period of reproduced oscillation and that of original one results in large L_2 errors for oscillating systems. We here use the amplitude and the frequency of oscillation to evaluate the similarity between the reproduced waveform and the original one. Hereafter, the error rate of the amplitude and the frequency are shown in the figures below for simplicity. The number of terms is also considered for the parsimonious model selection.

We consider two regression methods for SINDy, i.e. TLSA and Alasso, following our previous tests. Let us present the results of parameter search utilizing TLSA in figure 6. Note here that we use 10 000 mapped vectors for the construction of a SINDy model.
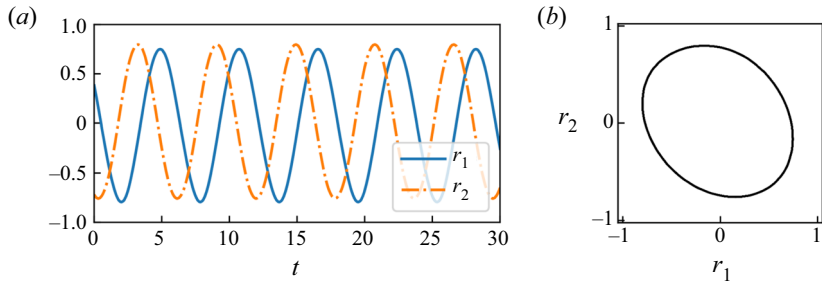
Figure 5. Latent vector dynamics of a periodic shedding case: (*a*) time history and (*b*) trajectory.

$\alpha = 1$

$$\frac{dx}{dt} = -31.47 + 46.50y + 111.51x^2 + 43.84xy + 113.30y^2 + 8.07x^3$$
$$- 159.21x^2y - 64.91xy^2 - 161.25y^3 - 99.11x^4 - 77.82x^3y$$
$$- 215.89x^2y^2 - 78.97xy^3 - 101.88y^4 - 13.49x^5 + 134.12x^4y$$
$$+ 100.88x^3y^2 + 306.65x^2y^3 + 115.73xy^4 + 143.17y^5$$

$$\frac{dy}{dt} = 46.21 - 65.21x - 47.33y - 156.75x^2 - 70.60xy - 165.10y^2$$
$$+ 213.25x^3 + 266.52x^2y + 287.76xy^2 + 170.79y^3 - 133.16x^4$$
$$+ 119.23x^3y + 307.13x^2y^2 + 125.57xy^3 + 147.49y^4 - 177.03x^5$$
$$- 313.64x^4y - 524.59x^3y^2 - 497.88x^2y^3 - 311.64xy^4 - 154.68y^5$$

$\alpha = 10$

$$\frac{dx}{dt} = 36.00y - 120.79x^2y - 48.51xy^2 - 121.02y^3 + 103.95x^4y$$
$$+ 87.83x^3y^2 + 228.10x^2y^3 + 84.11xy^4 + 104.53y^5$$

$$\frac{dy}{dt} = 46.21 - 65.21x - 47.33y - 156.75x^2 - 70.60xy - 165.10y^2$$
$$+ 213.25x^3 + 266.52x^2y + 287.76xy^2 + 170.79y^3 + 133.16x^4$$
$$+ 119.23x^3y + 307.13x^2y^2 + 125.57xy^3 + 147.49y^4 - 177.03x^5$$
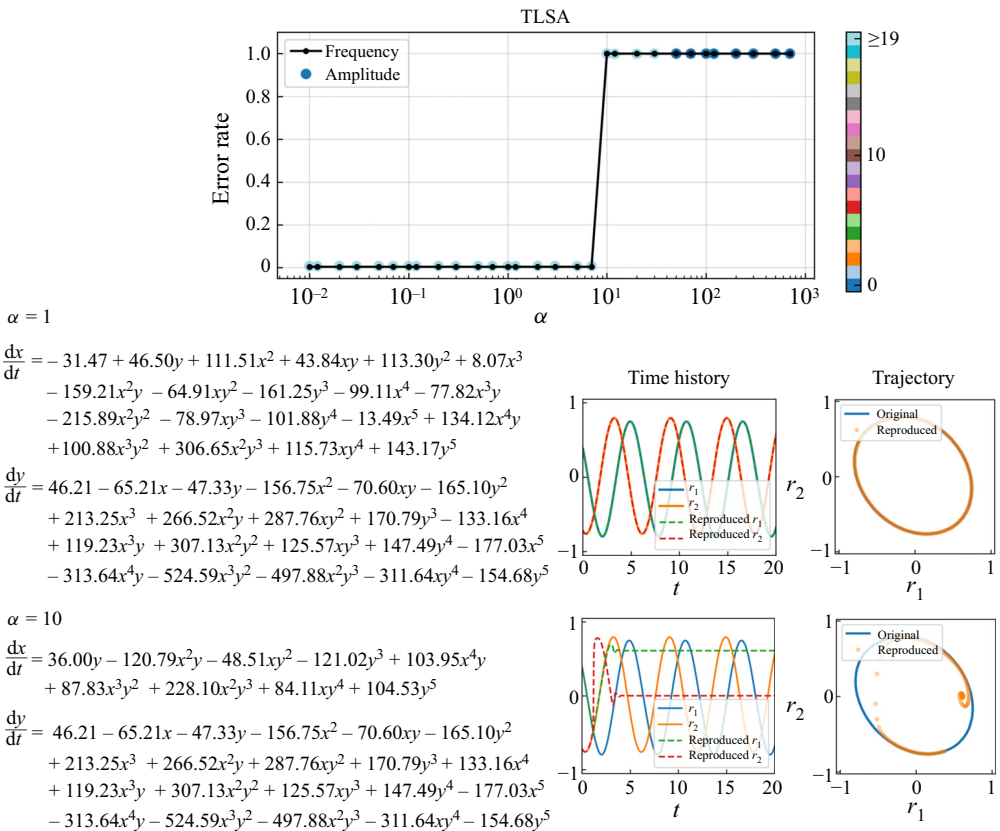$$- 313.64x^4y - 524.59x^3y^2 - 497.88x^2y^3 - 311.64xy^4 - 154.68y^5$$

Figure 6. Results with TLSA for the periodic cylinder example. Parameter search results, examples of obtained equations, and reproduced trajectories with $\alpha = 1$ and $\alpha = 10$ are shown. Colour of amplitude plot indicates the total number of terms. In the ordinary differential equations, the latent vector components (r_1, r_2) are represented by (x, y) for clarity.

Using $\alpha = 1$, the reproduced trajectory is in excellent agreement with the original one. However, there are many terms in the ordinary differential equations and some coefficients are too large, although the oscillation are presented with a shallow range $(-1, 1)$. The result with the TLSA here is likely because we do not use the data processing, which causes the large regression coefficients and non-sparse results. On the other hand, the reproduced trajectory with $\alpha = 10$ converges to a point around $(r_1, r_2) = (0.6, 0)$, although the model
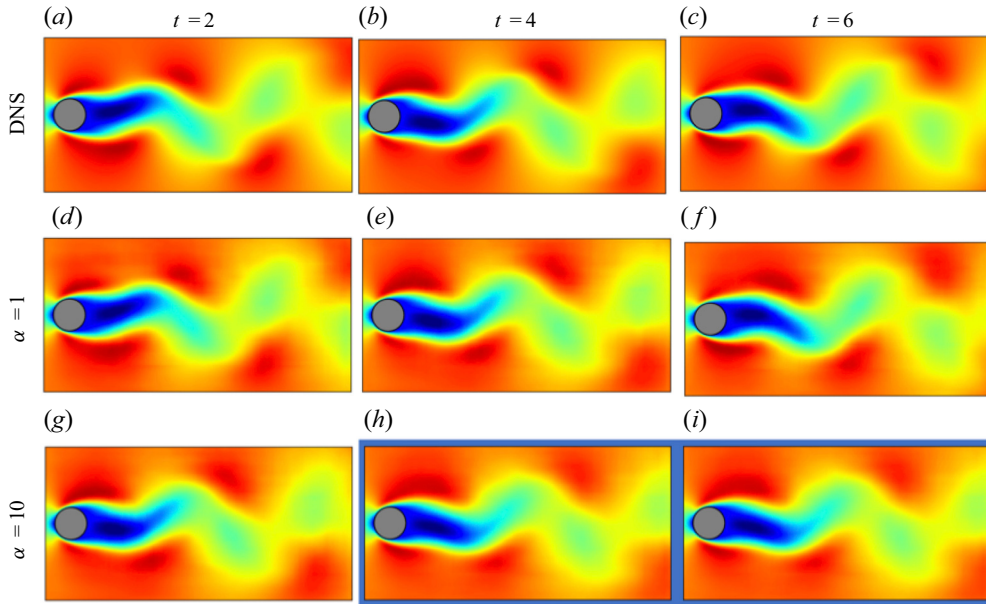
926 A10-8

Figure 7. Temporally evolved flow fields of DNS and the reproduced flow field with $\alpha = 1$ and $\alpha = 10$. With $\alpha = 10$, the flow field is frozen after around $t = 4$ (enhanced using blue colour).

becomes parsimonious. Since the predicted latent vector converges after $t \approx 4$ as shown in the time history of figure 6, the temporal evolution of the flow field by the CNN decoder is also frozen, as shown in figure 7. Other observation here is that there remains no terms in the ordinary differential equations with high threshold, i.e. $\alpha \geq 50$.

With Alasso as the regression method, we can find the candidate models with low errors at $\alpha = 2 \times 10^{-5}$ and 1×10^{-4}, as shown in figure 8. At $\alpha = 2 \times 10^{-5}$, the ordinary differential equations consist of four coefficients in total and the reproduced trajectory gradually converges as shown. On the other hand, by choosing the appropriate sparsity constant, i.e. $\alpha = 1 \times 10^{-4}$, the circle-like oscillating trajectory, which is similar to the reference, can be represented with only two terms. We then check the reproduced flow field by the combination of the predicted latent vector by SINDy and CNN decoder, as shown in figure 9. The temporal evolution of high-dimensional dynamics can be reproduced well using the proposed model, although the period of the two fields are slightly different.

The quantitative analysis with the probability density function, the mean streamwise velocity at $y = 0$, and the time history of ERR is summarized in figure 10. The ERR $\mathcal{R}$ is defined as

$$\mathcal{R} = \frac{\int_S (u'^2_{Rep} + v'^2_{Rep})\, dS}{\int_S (u'^2_{DNS} + v'^2_{DNS})\, dS}, \tag{3.3}$$

where (u'_{Rep}, v'_{Rep}) and (u'_{DNS}, v'_{DNS}) denote the fluctuation components of the reproduced velocity and the original velocity, respectively. For the probability density function, the distribution of the CNN-SINDy model is in excellent agreement with the reference DNS data. Since the SINDy model of the present case integrates the latent vector via the CNN-AE which can map a high-dimensional flow field into low-dimensional space efficiently thanks to the nonlinear activation function (Murata *et al.* 2020), the SINDy outperforms POD with two modes as long as the appropriate waveforms can be obtained.
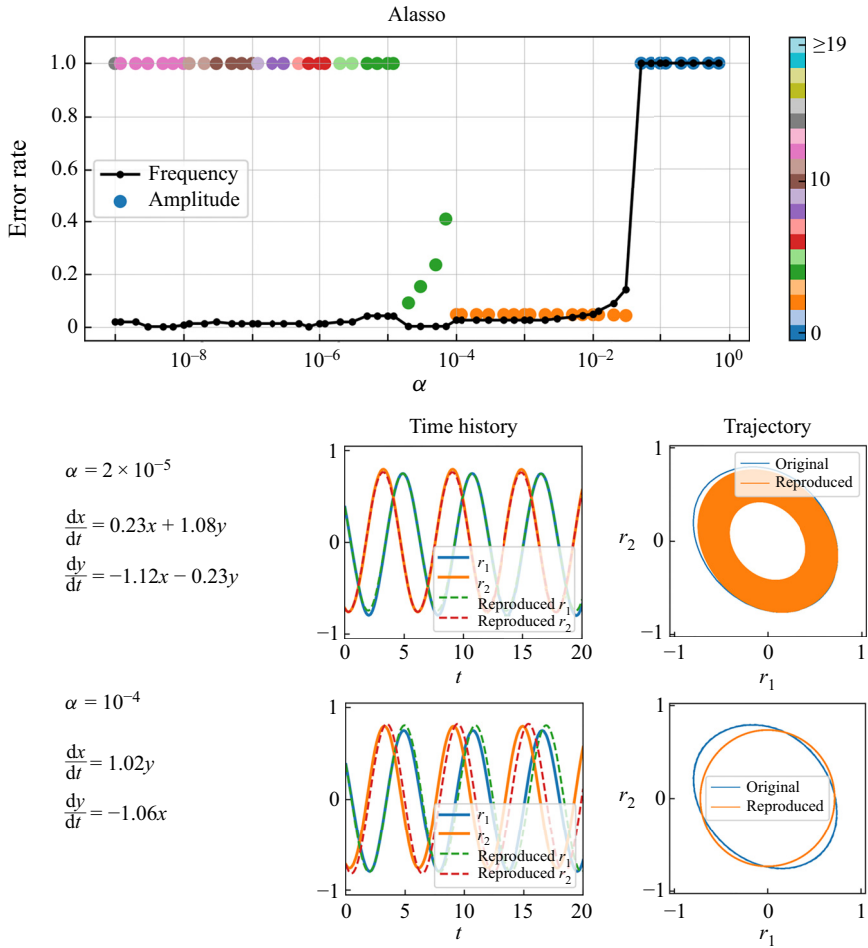
Figure 8. Results with Alasso of a periodic cylinder example. Parameter search results, examples of obtained equations, and reproduced trajectories with $\alpha = 2 \times 10^{-5}$ and $\alpha = 1 \times 10^{-4}$ are shown. Colour of amplitude plot indicates the total number of terms. In the ordinary differential equations, the latent vector components (r_1, r_2) are represented by (x, y) for clarity.

In other words, the obtained equation can be integrated stably. The success of the CNN-SINDy-based modelling can be also seen with time-averaged velocity and energy containing rate in figure 10.

3.2. *Example 2: transient wake of cylinder flow*

As the second example for the combination of CNN and SINDy, we consider the transient process of a flow around a circular cylinder at $Re_D = 100$. For taking into account the transient process, the streamwise length of the computational domain and that of the flow field data are extended to $L_x = 51.2$ and $8.2 \le x \le 37$, i.e. $N_x^* = 1152$, same set-up with Murata *et al.* (2020). The time history and the trajectory in the latent space are shown in figure 11. Since the trajectory looks like a circle as shown in figure 11(*b*), we use the residual sum of error of $r_1^2 + r_2^2$ between the original value and the reproduced value as the evaluation index in this example.
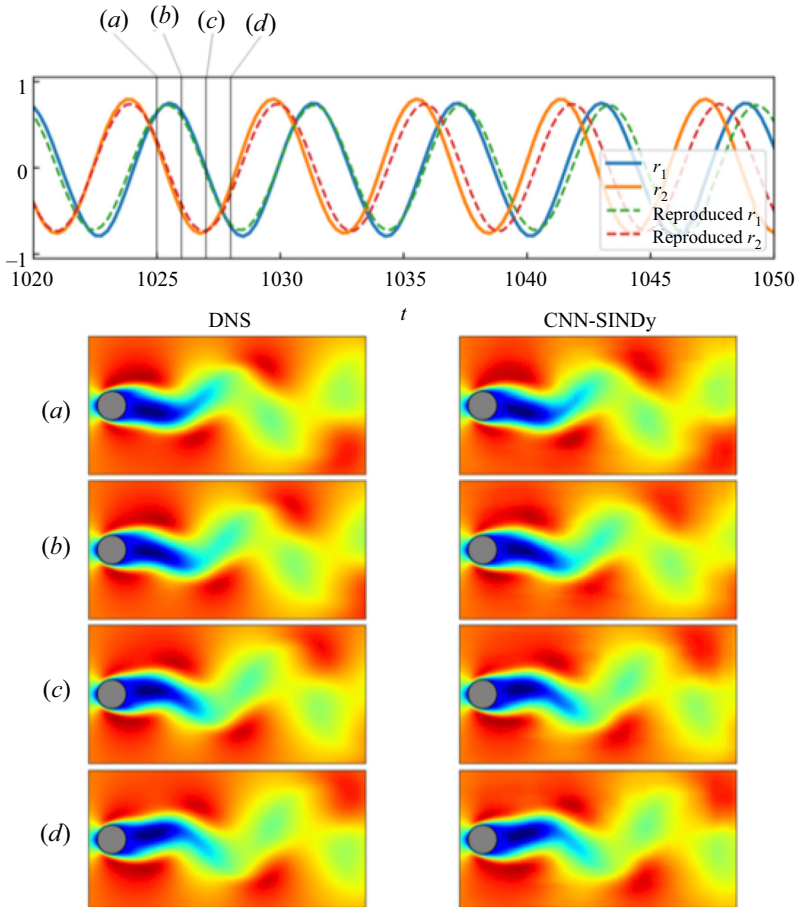
926 A10-10

Figure 9. Time history of the latent vector and temporal evolution of the wake of DNS and the reproduced field at (*a*) $t = 1025$, (*b*) $t = 1026$, (*c*) $t = 1027$ and (*d*) $t = 1028$.

For SINDy, we use the data with $t = [60, 160]$. The library matrix contains up to fifth-order terms. Although equations, which can provide the correct temporal evolution of the latent vector, can be obtained with Alasso by including higher-order terms, the model here is, of course, not sparse and so difficult to interpret.

The parameter search results with TLSA for transient wake are summarized in figure 12. Since the trajectory here is simple as shown, we can obtain the governing equations, which provide the reasonable agreement with the reference, at $\alpha = 7 \times 10^{-2}$. With a higher sparsity constant, i.e. $\alpha = 0.7$, the equation provides a temporal evolution of the latent vector with almost no oscillation.

Alasso is also considered, as shown in figure 13. Although the number of terms is larger than that in the periodic shedding example, the trajectory can be reproduced successfully at $\alpha = 1.2 \times 10^{-8}$. With a higher sparsity constant, e.g. $\alpha = 1.2 \times 10^{-2}$, the developing oscillation is not reproduced due to a lack of number of terms. Noteworthy here is that the error for frequency is almost negligible in the range of $5 \times 10^{-6} \leq \alpha \leq 10^{-3}$ in spite of the high error rate regarding $x^2 + y^2$. This is because the slight oscillation occurs with a similar frequency to the solution. Although the identified equation here is more complex than the well known transient system with a two-dimensional paraboloic manifold (Sipp
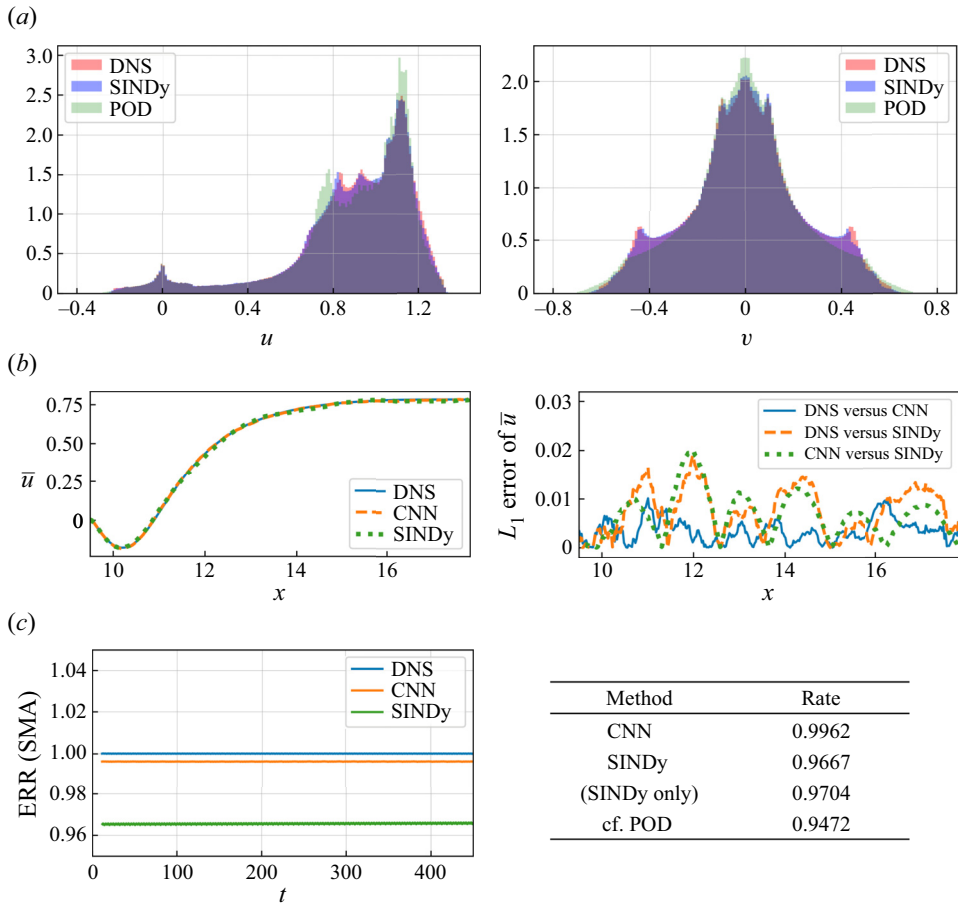
926 A10-11

(*a*)

(*b*)

(*c*)

Method	Rate
CNN	0.9962
SINDy	0.9667
(SINDy only)	0.9704
cf. POD	0.9472

Figure 10. Evaluation of reproduced flow field with a periodic shedding. (*a*) Probability density function, (*b*) mean streamwise velocity at $y = 0$ and (*c*) the energy reconstruction rate (ERR). Simple moving average (SMA) of ERR is shown here for the clearness.

(*a*)

(*b*)
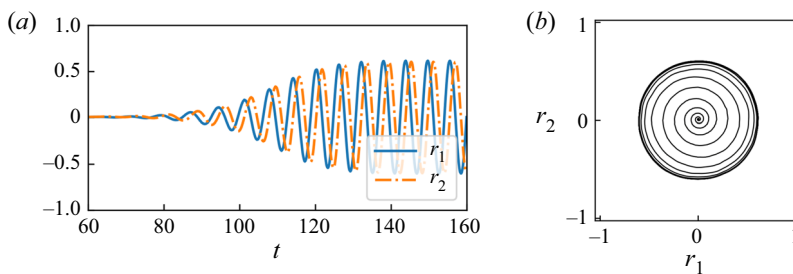
Figure 11. Latent vector dynamics of the transient process: (*a*) time history and (*b*) trajectory.

& Lebedev 2007; Loiseau & Brunton 2018; Loiseau, Brunton & Noack 2018*a*), it is likely caused by the complexity of captured information inside the AE modes compared with the POD modes (Murata *et al.* 2020). This observation enables us to notice the trade-off relationship between the used information associated with the number of modes and the sparsity of equation. Hence, taking a constraint for nonlinear AE modes to be useful, e.g.

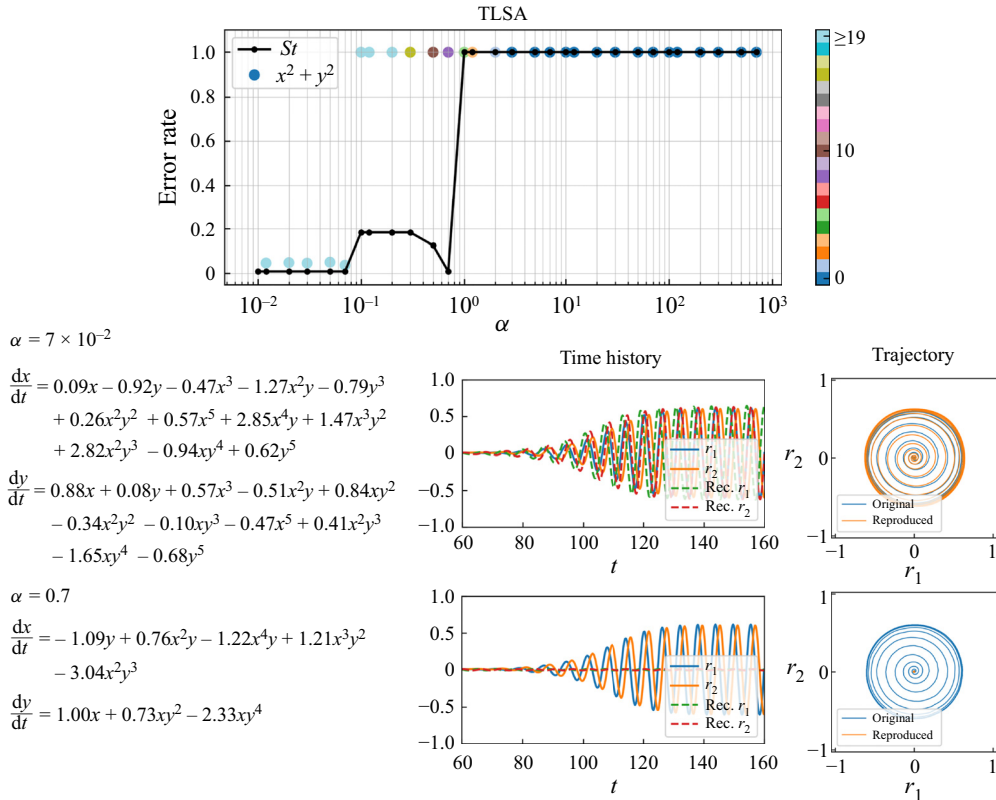Figure 12. Results with TLSA of the transient example. Parameter search results, examples of obtained equations, and reproduced trajectories with $\alpha = 7 \times 10^{-2}$ and $\alpha = 0.7$ are shown. Colour of amplitude plot indicates the total number of terms. In the ordinary differential equations, the latent vector components (r_1, r_2) are represented by (x, y) for clarity.

orthogonal, may be helpful in identifying more sparse dynamics (Champion *et al.* 2019*a*; Ladjal, Newson & Pham 2019; Jayaraman & Mamun 2020).

The streamwise velocity fields reproduced by the CNN-AE and the integrated latent variables with the Alasso-based SINDy are presented in figure 14. The present two sparsity constants here correspond to the results in figure 13. With $\alpha = 1.2 \times 10^{-8}$, the flow field can successfully be reproduced thanks to nonlinear terms in the identified equations by SINDy. In contrast, it is tough to capture the transient behaviour correctly by the CNN-SINDy model with $\alpha = 1.2 \times 10^{-2}$, which can explicitly be found by the comparison among (*b*) (where $t = 80$) in figure 14. This corresponds to the observation that the sparse model cannot reproduce the trajectory well in figure 13.

3.3. *Example 3: two-parallel cylinders wake*

To demonstrate the applicability of the present CNN-SINDy-based ROM to more complex wake flows, let us here consider a flow around two-parallel cylinders whose radii are different from each other (Morimoto *et al.* 2020*b*, 2021), as presented in figure 1. Unlike the wake dynamics of two identical circular cylinders as described in Kang (2003), the coupled wakes behind two side-by-side uneven cylinders exhibits more complex vortex interactions, due to the mismatch in their individual shedding frequencies.

$\alpha = 1.2 \times 10^{-8}$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 0.06x - 0.92y - 0.19x^3 - 1.29x^2y$$
$$- 0.78y^3 + 2.91x^4y + 1.14x^3y^2 + 2.84x^2y^3$$
$$- 0.72xy^4 + 0.58y^5$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = 0.88x + 0.08y + 0.55x^3 - 0.50x^2y$$
$$+ 0.85xy^2 - 0.34x^2y^2 - 0.43x^5 + 0.37x^2y^3$$
$$- 1.69xy^4 - 0.65y^5$$

$\alpha = 1.2 \times 10^{-2}$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -0.74y$$
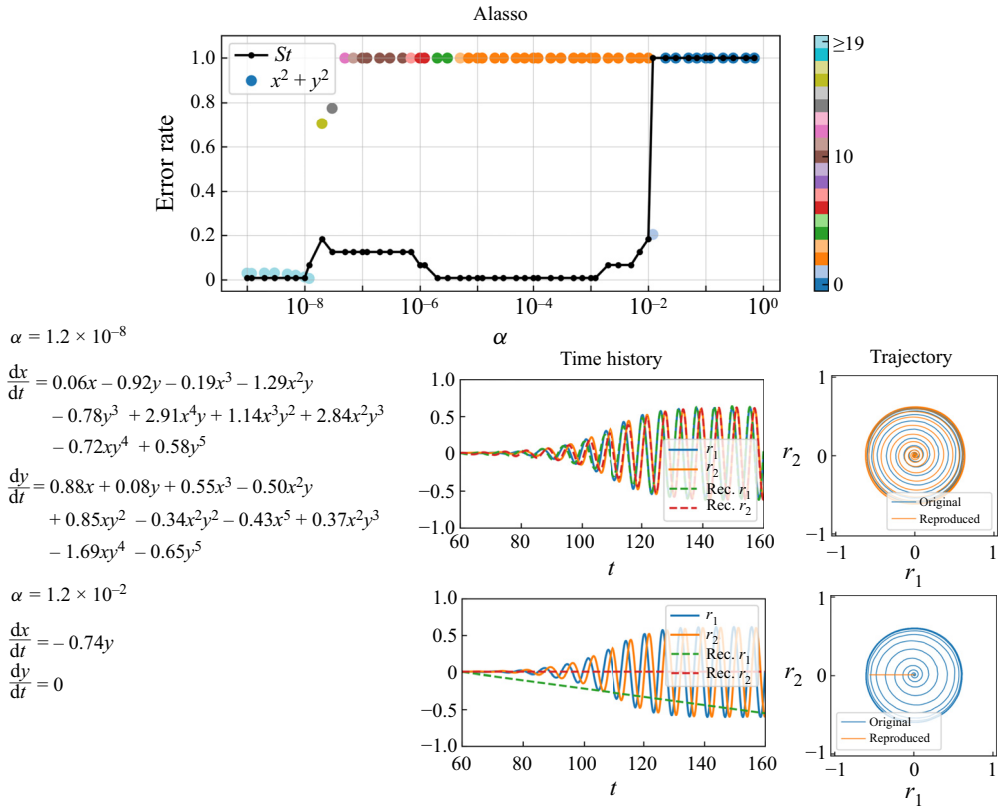$$\frac{\mathrm{d}y}{\mathrm{d}t} = 0$$

Figure 13. Results with Alasso of the transient example. Parameter search results, examples of obtained equations, and reproduced trajectories with $\alpha = 1.2 \times 10^{-8}$ and $\alpha = 1.2 \times 10^{-2}$ are shown. Colour of amplitude plot indicates the total number of terms. In the ordinary differential equations, the latent vector components (r_1, r_2) are represented by (x, y) for clarity.
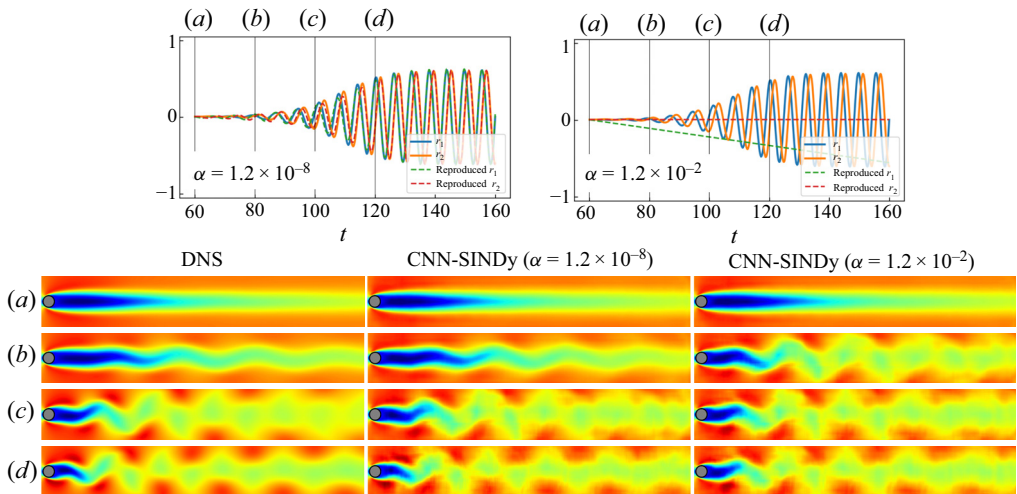

Figure 14. Time history of latent vector and the temporal evolution of the wake of DNS and the reproduced field at (*a*) $t = 60$, (*b*) $t = 80$, (*c*) $t = 100$ and (*d*) $t = 120$.
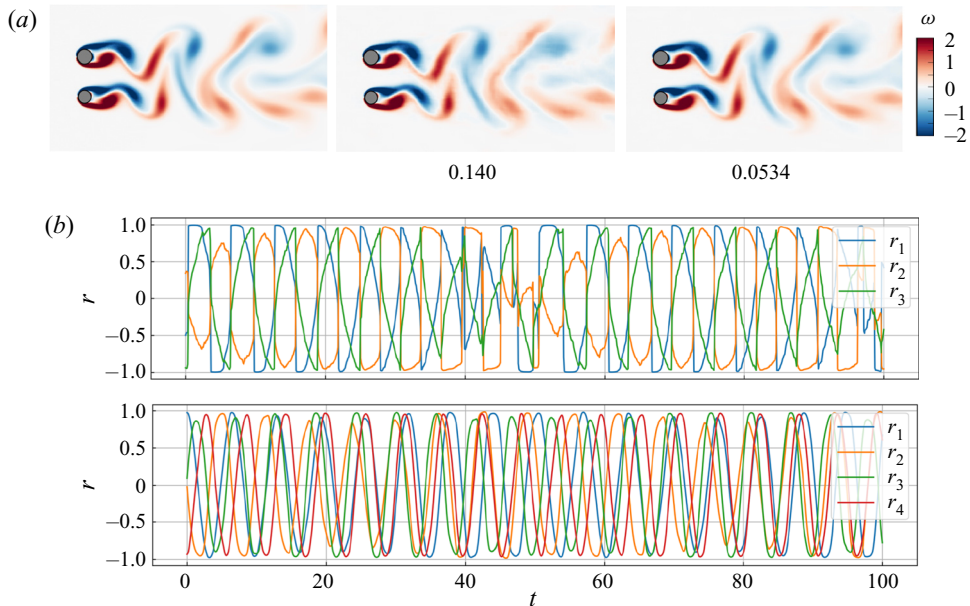
926 A10-14

Figure 15. Autoencoder-based low-dimensionalization for the wake of the two-parallel cylinders. (*a*) Comparison of the reference DNS, the decoded field with $n_r = 3$ and the decoded field with $n_r = 4$. The values underneath the contours indicate the L_2 error norm. (*b*) Time series of the latent variables with $n_r = 3$ and 4.

Training flow snapshots are prepared with a DNS with the open-source computational fluid dynamics (known as CFD) toolbox OpenFOAM (Weller *et al.* 1998), using second-order discretization schemes in both time and space. We arrange the two circular cylinders with a size ratio of *r* and a gap of *gD*, where *g* is the gap ratio and *D* is the diameter of the lower cylinder. The Reynolds number is fixed at $Re_D = 100$. The two cylinders are placed 20*D* downstream of the inlet where a uniform flow with velocity U_∞ is prescribed, and 40*D* upstream of the outlet with zero pressure. The side boundaries are specified as slip and are 40*D* apart. The time steps for the DNS and the snapshot sampling are, respectively, 0.01 and 0.1. A notable feature of wakes associated with the present two-cylinder position is complex wake behaviour caused by varying the size ratio *r* and the gap ratio *g* (Morimoto *et al.* 2020*b*, 2021). In the present study, the wake with the combination of $\{r, g\} = \{1.15, 2.0\}$, which is a quasi-periodic in time, is considered for the demonstration. For the training of both the CNN-AE and the SINDy, we use 2000 snapshots. The vorticity field ω is used as a target attribute in this example.

Prior to the use of SINDy, let us determine the size of latent variables n_r of this example, as summarized in figure 15. We here construct AEs with two n_r cases: $n_r = 3$ and $n_r = 4$. As presented in figure 15(*a*), the recovered flow fields through the AE-based low-dimensionalization are in reasonable agreement with the reference DNS snapshots. Quantitatively speaking, the L_2 error norms between the reference and the decoded fields are approximately 14 % and 5 % for each case. Although both models achieve the smooth spatial reconstruction, we can find the significant difference by focusing on their temporal behaviours in figure 15(*b*). While the time trace with $n_r = 4$ shows smooth variations for all four variables, the curves with $n_r = 3$ exhibit a shaky behaviour despite the quasi-periodic nature of the flow. This is caused by the higher error level of the $n_r = 3$ case. Furthermore, it is also known that a high-error level due to an overcompression via
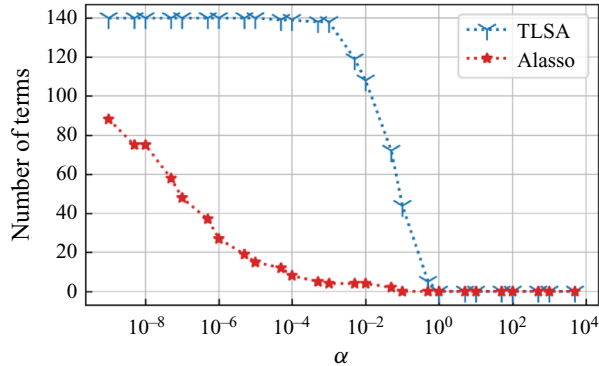
926 A10-15

Figure 16. The relationship between the sparsity constant α and the number of terms in identified equations via SINDys with TLSA and Alasso for the two-parallel cylinder wake example.

AE is highly influential for a temporal prediction of them since there should be exposure bias (Endo, Tomobe & Yasuoka 2018) occurring by a temporal integration (Hasegawa *et al.* 2020*a*,*b*; Nakamura *et al.* 2021). In what follows, we use the AE with $n_r = 4$ for the SINDy-based ROM.

Let us perform SINDy with TLSA and Alasso for the AE latent variables with $n_r = 4$. For the construction of a library matrix of this example, we include up to third-order terms in terms of four latent variables. The relationship between the sparsity constant α and the number of terms in identified equations is examined in figure 16. Similarly to the previous two examples with a single cylinder, the trends for TLSA and Alasso are different from each other. Note that one can only refer to this map to check the sparsity trend of the identified equation, since there is no model equation. To validate the fidelity of the equations, we need to integrate identified equations and decode a flow field using the decoder part of AE.

The identified equations via the TLSA-based SINDy are investigated in figure 17. As examples, the cases with $\alpha = 0.1$ and 0.5 are shown. The case with $\alpha = 1$ which contains nonlinear terms shows a diverging behaviour at $t \approx 90$. The sparse equation obtained with $\alpha = 0.5$ can provide stable solutions for r_1 and r_2; however, its sparseness causes the unstable integration for r_3 and r_4. Note in passing that the TLSA-based modelling cannot identify the stable solution although we have carefully examined the influence on the sparsity factors for other cases not shown here.

We then use Alasso as a regression function of SINDy, as summarized in figure 18. The cases with $\alpha = 5 \times 10^{-7}$ and 10^{-6} are shown. The identified equation with $\alpha = 5 \times 10^{-7}$ can successfully be integrated and its temporal behaviour is in good agreement with the reference curve provided by the AE-based latent variables. However, what is notable here is that the equation with $\alpha = 10^{-6}$ shows the decaying behaviour with an increasing of the integration window despite the equation form being very similar to the case with $\alpha = 5 \times 10^{-7}$. Some nonlinear terms are eliminated due to the slight increase in the sparsity constant. These results indicate the importance of the careful choice for both regression functions and the sparsity constants associated with them.

Based on the results above, the integrated variables with the case of Alasso with $\alpha = 5 \times 10^{-7}$ are fed into the CNN decoder, as presented in figure 19. The reproduced flow fields are in reasonable agreement with the reference DNS data, although there is a slight offset due to numerical integration. Hence, the present reduced-order model is able to achieve a reasonable wake reconstruction by following only the temporal evolution of

926 A10-16

(*a*)

$$\dot{r}_1 = 0.410r_1 + 0.533r_2 - 0.265r_1^2 r_3 + 0.105r_1^2 r_4$$
$$+ 0.420r_1 r_2 r_3 - 0.414r_1 r_3^2 + 0.165r_1 r_3 r_4 - 0.415r_1 r_4^2 +$$
$$0.631r_2^3 - 0.225r_2^2 r_3 + 0.441r_3^3 + 0.301r_3 r_4^2$$

$$\dot{r}_2 = -0.397r_1 - 0.104r_2 + 0.338r_4 - 0.764r_1^3 - 0.232r_1^2 r_4 -$$
$$-0.262r_1 r_2 r_3 - 0.142r_1 r_3^2 + 0.173r_2^3 - 0.266r_2^2 r_3 -$$
$$0.309r_2^2 r_4 - 0.193r_2 r_3^2$$

$$\dot{r}_3 = -0.725r_4 - 0.141r_1^3 - 0.267r_1^2 r_4 - 0.157r_1 r_2^2 -$$
$$0.140r_1 r_2 r_3 + 0.155r_1 r_3^2 - 0.152r_1 r_3 r_4 + 0.101r_1 r_4^2 +$$
$$0.111r_2^2 r_3 - 0.355r_2^2 r_4 + 0.431r_3^2 r_4 + 0.156r_3 r_4^2 - 0.338r_4^3$$

$$\dot{r}_4 = 0.792r_3 - 0.213r_1 r_2 r_3 - 0.161r_1 r_3 r_4 - 0.164r_2^3 +$$
$$+ 0.147r_2 r_3^2 + 0.407r_3^3 - 0.147r_3^2 r_4 - 0.114r_4^3$$

(*b*)

$$\dot{r}_1 = -0.984r_2$$

$$\dot{r}_2 = -1.29r_1^3$$

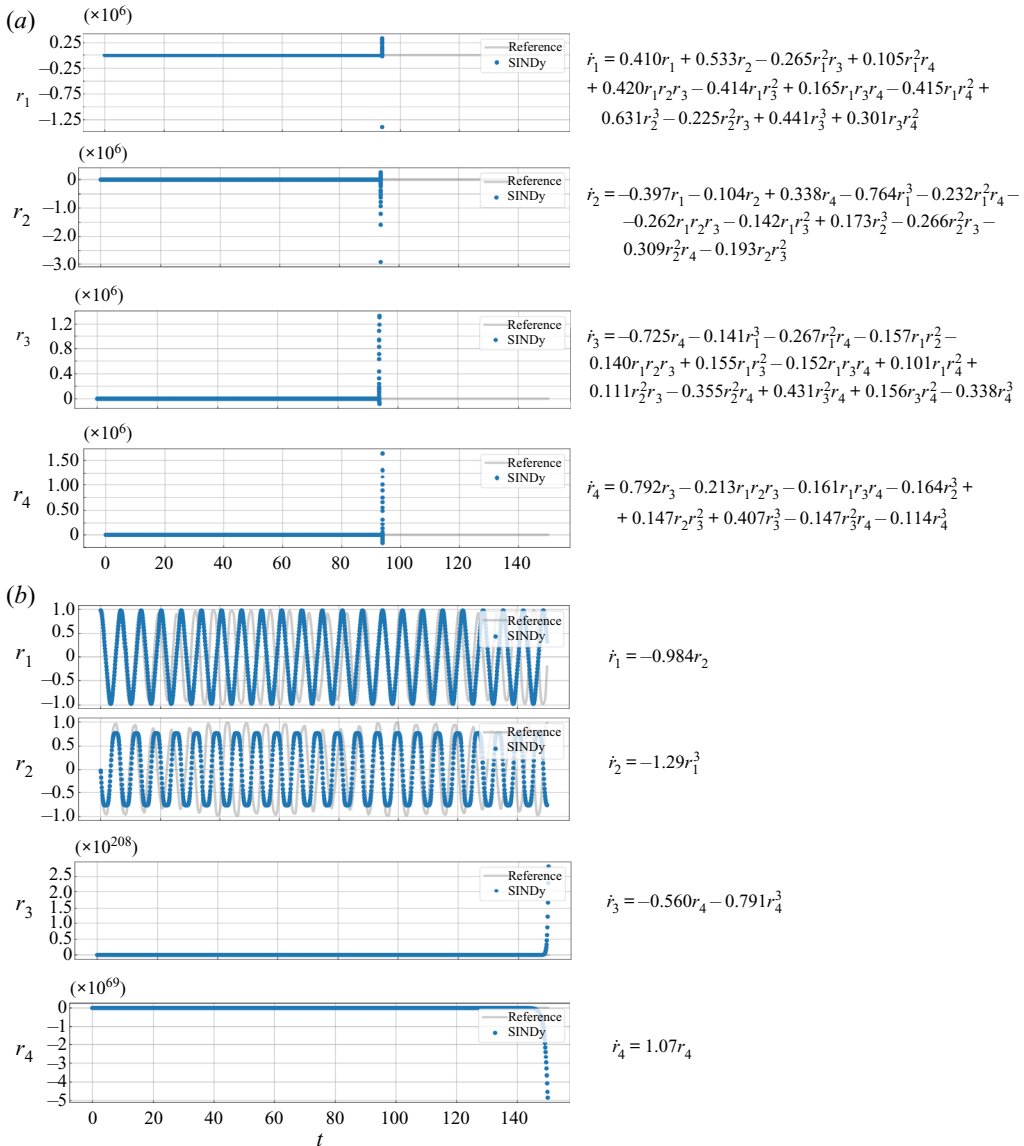$$\dot{r}_3 = -0.560r_4 - 0.791r_4^3$$

$$\dot{r}_4 = 1.07r_4$$

Figure 17. Integration of the identified equations via the TLSA-based SINDy for the two-parallel cylinder wake example. The cases are (*a*) TLSA, $\alpha = 0.1$ and (*b*) TLSA, $\alpha = 0.5$.

low-dimensionalized vector and caring the selection of parameters, which is akin to the observation of single cylinder cases.

3.4. *Outlook: nine-equation shear flow model*

One of the remaining issues of the present CNN-SINDy model is the applicability to flows where a lot of spatial modes are required to reconstruct the flow, e.g. turbulence. With the current scheme for the CNN-AE, it is difficult to compress turbulent flow data while keeping the information of high-dimensional dynamics (Murata *et al.* 2020). We have also recently reported that the difficulty for turbulence low-dimensionalization still remains

(a)

$$\dot{r}_1 = 0.339r_1 + 0.563r_2 - 0.215r_1^2r_3 + 0.424r_1r_2r_3$$
$$- 0.313r_1r_3^2 - 0.158r_1r_3r_4 - 0.360r_1r_4^2 + 0.613r_2^3 -$$
$$0.162r_2^2r_3 + 0.388r_3^3 + 0.206r_3r_4^2 + 0.135r_4^3$$

$$\dot{r}_2 = -0.395r_1 + 0.285r_4 - 0.752r_1^3 - 0.175r_1^2r_4$$
$$- 0.281r_1r_2r_3 - 0.156r_1r_3^2 - 0.282r_2^2r_3$$
$$- 0.252r_2^2r_4 - 0.178r_2r_3^2$$

$$\dot{r}_3 = -0.796r_4 + 0.111r_2^2r_3 + 0.106r_2r_3r_4$$
$$- 0.0759r_2r_4^2 + 0.233r_3^2r_4 + 0.248r_3r_4^2 - 0.604r_4^3$$

$$\dot{r}_4 = 0.743r_3 + 0.0897r_1^2r_3 - 0.201r_1r_2r_3 - 0.163r_1r_3r_4$$
$$- 0.172r_2^3 + 0.167r_2r_3^2 + 0.414r_3^3 - 0.138r_3^2r_3 - 0.114r_4^3$$

(b)

$$\dot{r}_1 = 0.579r_2 - 0.103r_1^2r_3 + 0.425r_1r_2r_3$$
$$+ 0.139r_1r_3r_4 + 0.615r_2^3 + 0.272r_3^3 + 0.144r_4^3$$

$$\dot{r}_2 = -0.358r_1 + 0.0703r_4 - 0.776r_1^3$$
$$- 0.261r_1r_2r_3 - 0.172r_1r_3^2 - 0.284r_2^2r_3 - 0.175r_2r_3^2$$

$$\dot{r}_3 = -1.08r_4 + 0.129r_2^2r_3$$
$$+ 0.507r_3^2r_4 + 0.146r_3r_4^2 - 0.291r_4^3$$

$$\dot{r}_4 = 0.791r_3 - 0.210r_1r_2r_3 - 0.158r_1r_3r_4 - 0.156r_2^3$$
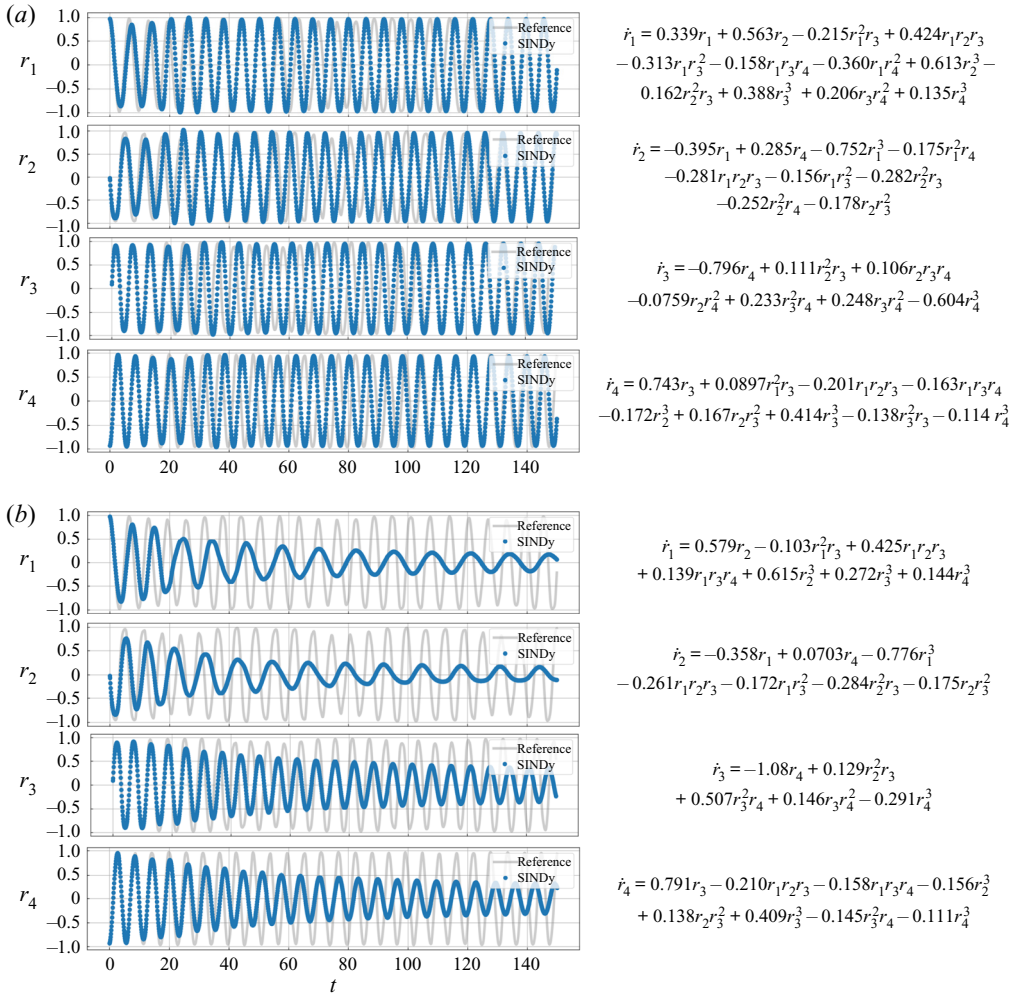$$+ 0.138r_2r_3^2 + 0.409r_3^3 - 0.145r_3^2r_4 - 0.111r_4^3$$

Figure 18. Integration of the identified equations via the Alasso-based SINDy for the two-parallel cylinder wake example. The cases are (a) Alasso, $\alpha = 5 \times 10^{-7}$ and (b) Alasso, $\alpha = 10^{-6}$.
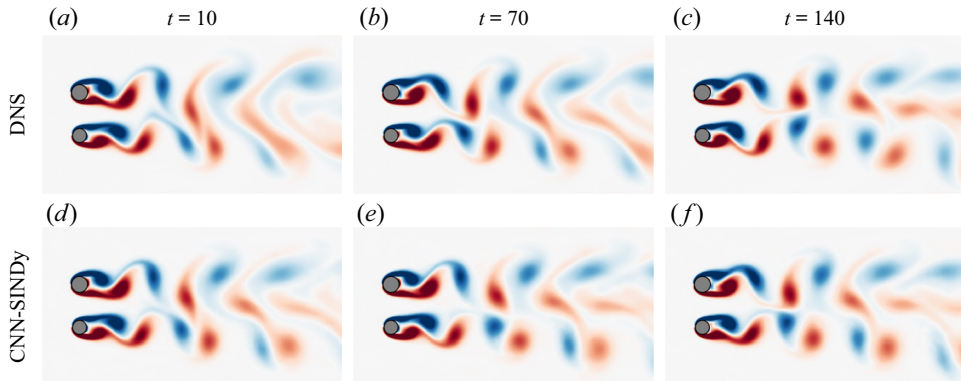

Figure 19. Reproduced fields with the CNN-SINDy-based ROM of the two-parallel cylinders example. The case of Alasso with $\alpha = 5 \times 10^{-7}$ is used for SINDy. The DNS flow fields are also shown for the comparison.

even if we use a customized AE, although it can achieve the better mapping than that by conventional AE and POD (Fukami, Nakamura & Fukagata 2020*c*). In addition, the number of terms in a coefficient matrix must be drastically increased for the turbulent case unless the flow can be expressed with a smaller number of modes. Hence, our next question here is 'can we also use SINDy if a turbulent flow can be mapped into a smaller number of modes?'. In this section, let us consider a nine-equation turbulent shear flow model between infinite parallel free-slip walls under a sinusoidal body force (Moehlis, Faisst & Eckhardt 2004) as the preliminary example for the application to turbulence with low number of modes.

In the nine-equation model, various statistics, including mean velocity profile, streaks and vortex structures, can be represented with only nine Fourier modes $u_j(x)$. Analogous to POD, the flow fields can be mathematically expressed with a superposition of temporal coefficients and modes such that

$$u(x, t) = \sum_{j=1}^{9} a_j(t) u_j(x). \tag{3.4}$$

Here, nine ordinary differential equations for the nine mode coefficients are as follows:

$$\frac{da_1}{dt} = \frac{\mu^2}{Re} - \frac{\mu^2}{Re}a_1 - \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\zeta\mu\gamma}}a_6 a_8 + \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\mu\gamma}}a_2 a_3, \tag{3.5}$$

$$\frac{da_2}{dt} = -Re^{-1}\left(\frac{4\mu^2}{3} + \gamma\right)a_2 + \frac{5\sqrt{2}}{3\sqrt{3}}\frac{\gamma^2}{\kappa_{\zeta\gamma}}a_4 a_6 - \frac{\gamma^2}{\sqrt{6}\kappa_{\zeta\gamma}}a_5 a_7 - \frac{\zeta\mu\gamma}{\sqrt{6}\kappa_{\zeta\gamma}\kappa_{\zeta\mu\gamma}}a_5 a_8$$

$$- \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\mu\gamma}}a_1 a_3 - \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\mu\gamma}}a_3 a_9, \tag{3.6}$$

$$\frac{da_3}{dt} = -\frac{\mu^2 + \gamma^2}{Re}a_3 + \frac{2}{\sqrt{6}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}}(a_4 a_7 + a_5 a_6)$$

$$+ \frac{\mu^2(3\zeta^2 + \gamma^2) - 3\gamma^2(\zeta^2 + \gamma^2)}{\sqrt{6}\kappa_{\zeta\gamma}\kappa_{\mu\gamma}\kappa_{\zeta\mu\gamma}}a_4 a_8, \tag{3.7}$$

$$\frac{da_4}{dt} = -\frac{3\zeta^2 + 4\mu^2}{3Re}a_4 - \frac{\zeta}{\sqrt{6}}a_1 a_5 - \frac{10}{3\sqrt{6}}\frac{\zeta^2}{\kappa_{\zeta\gamma}}a_2 a_6$$

$$- \sqrt{\frac{3}{2}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}}a_3 a_7 - \sqrt{\frac{3}{2}}\frac{\zeta^2\mu^2}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}\kappa_{\zeta\mu\gamma}}a_3 a_8 - \frac{\zeta}{\sqrt{6}}a_5 a_9, \tag{3.8}$$

$$\frac{da_5}{dt} = -\frac{\zeta^2 + \mu^2}{Re}a_5 + \frac{\zeta}{\sqrt{6}}a_1 a_4 + \frac{\zeta^2}{\sqrt{6}\kappa_{\zeta\gamma}}a_2 a_7$$

$$- \frac{\zeta\mu\gamma}{\sqrt{6}\kappa_{\zeta\gamma}\kappa_{\zeta\mu\gamma}}a_2 a_8 + \frac{\zeta}{\sqrt{6}}a_4 a_9 + \frac{2}{\sqrt{6}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}}a_3 a_6, \tag{3.9}$$

$$\frac{da_6}{dt} = -\frac{3\zeta^2 + 4\mu^2 + 3\gamma^2}{3Re}a_6 + \frac{\zeta}{\sqrt{6}}a_1 a_7 + \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\zeta\mu\gamma}}a_1 a_8$$

$$+ \frac{10}{3\sqrt{6}}\frac{\zeta^2 - \gamma^2}{\kappa_{\zeta\gamma}}a_2 a_4 - 2\sqrt{\frac{2}{3}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}}a_3 a_5 + \frac{\zeta}{\sqrt{6}}a_7 a_9 + \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\zeta\mu\gamma}}a_8 a_9, \tag{3.10}$$

926 A10-19

$$\frac{\mathrm{d}a_7}{\mathrm{d}t} = -\frac{\zeta^2 + \mu^2 + \gamma^2}{Re}a_7 - \frac{\zeta}{\sqrt{6}}(a_1a_6 + a_6a_9)$$

$$+ \frac{1}{\sqrt{6}}\frac{\gamma^2 - \zeta^2}{\kappa_{\zeta\gamma}}a_2a_5 + \frac{1}{\sqrt{6}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\mu\gamma}}a_3a_4, \tag{3.11}$$

$$\frac{\mathrm{d}a_8}{\mathrm{d}t} = -\frac{\zeta^2 + \mu^2 + \gamma^2}{Re}a_8 + \frac{2}{\sqrt{6}}\frac{\zeta\mu\gamma}{\kappa_{\zeta\gamma}\kappa_{\zeta\mu\gamma}}a_2a_5 + \frac{\gamma^2(3\zeta^2 - \mu^2 + 3\gamma^2)}{\sqrt{6}\kappa_{\zeta\gamma}\kappa_{\mu\gamma}\kappa_{\zeta\mu\gamma}}a_3a_4, \tag{3.12}$$

$$\frac{\mathrm{d}a_9}{\mathrm{d}t} = -\frac{9\mu^2}{Re}a_9 + \sqrt{\frac{3}{2}}\frac{\mu\gamma}{\kappa_{\mu\gamma}}a_2a_3 - \frac{\mu\gamma}{\kappa_{\zeta\mu\gamma}}a_6a_8, \tag{3.13}$$

where ζ, μ and γ are constant values, $\kappa_{\zeta\gamma} = \sqrt{\zeta^2 + \gamma^2}$, $\kappa_{\mu\gamma} = \sqrt{\mu^2 + \gamma^2}$, $\kappa_{\zeta\mu\gamma} = \sqrt{\zeta^2 + \mu^2 + \gamma^2}$. These coefficients are multiplied to corresponding Fourier modes which have individual roles in reconstructing a flow, e.g. basic profile, streak and spanwise flows. We refer enthusiastic readers to Moehlis *et al.* (2004) for details.

In this section, we aim to obtain the coefficient matrix for the simultaneous time differential equations (3.5) to (3.13) using SINDy. In the following, let us consider the Reynolds number based on the channel half-height δ and laminar velocity U_0 at a distance of $\delta/2$ from the top wall set to $Re = 400$. The initial condition for numerical integration of the equations above is $(a_1^0, a_1^0, a_2^0, a_3^0, a_4^0, a_5^0, a_6^0, a_7^0, a_8^0, a_9^0) = (1, 0.07066, -0.07076, 0, 0, 0, 0, 0, 0)$ with a random small perturbation for a_4, which is the same set up as the reference code by Srinivasan *et al.* (2019). The lengths and the number of grids of the computational domain are set to $(L_x, L_y, L_z) = (4\pi, 2, 2\pi)$ and $(N_x, N_y, N_z) = (21, 21, 21)$, respectively. The constant values are set to $(\zeta, \mu, \gamma) = (2\pi/L_x, \pi/2, 2\pi/L_z)$ and the time step is 0.5. Examples of streamwise-averaged velocity u_x contour and velocity u_y contour at the midplane with the temporal evolution of the amplitudes a_i and the pairwise correlations of the present nine coefficients are shown in figure 20. The chaotic nature of the considered problem can be seen. For performing SINDy, we use 10 000 discretized coefficients as the training data.

The SINDy in this section is also performed with TLSA and Alasso following the discussions above. Since the equations for the temporal coefficients are constructed up to second-order terms, the coefficient matrix also includes up to second-order terms, as shown in figure 21(*a*). The total number of terms considered here is 55. The results with TLSA and Alasso are summarized in figure 21(*b*). The matrices located in the right-hand area correspond to the coefficient matrix in figure 21(*a*). Similar to the results above, the model using TLSA has some huge values due to a lack of penalty terms. Especially, the effects by overfitting can be seen for the low-order portion. On the other hand, the remarkable ability of the SINDy can be seen with the Alasso. By giving the appropriate sparsity constant α, the governing equations can be represented successfully. The details of each magnitude of the coefficients are shown in figure 22. It is striking that the dominant terms are perfectly captured by using SINDy, although the magnitudes are slightly different. These noteworthy results indicate that a governing equation of low-dimensionalized turbulent flows can be obtained from only time series data by using SINDy with appropriate parameter selections. In other words, we may also be able to construct a machine-learning-based reduced-order model for turbulent flows with the interpretable sense as a form of equation, if a well-designed model for mapping into low-dimensional manifolds can be constructed.

The robustness of SINDy against noisy measurements observed in the training pipeline for this example is finally assessed in figure 23. We here introduce the Gaussian
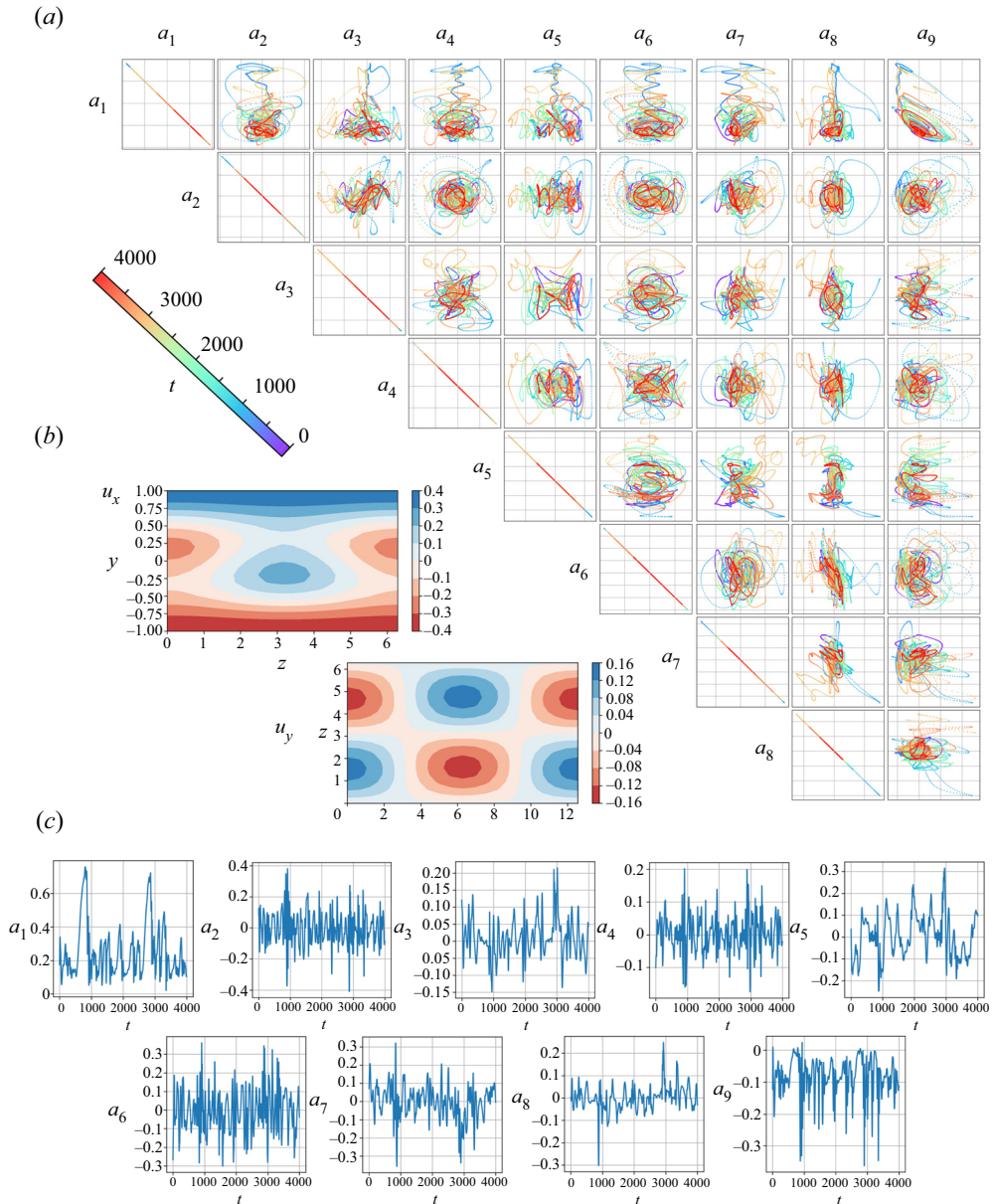
Figure 20. (*a*) Pairwise correlations of nine coefficients. (*b*) Example contours of velocity u_x and velocity u_y at midplane. (*c*) Temporal evolution of amplitudes a_i.

perturbation for the training data as

$$f_m = f + \gamma_m n, \tag{3.14}$$

where f is a target vector, γ is the magnitude of the noise and n denotes the Gaussian noise. As the magnitude of noise, four cases $\gamma_m = \{0.05, 0.10, 0.20, 0.40\}$ are considered. With $\gamma = 0.05$, SINDy is still be able to identify the equation accordingly. However, the first-order terms are eliminated and unnecessary high-order terms are added with an increasing of the noise magnitude of the training data, although the whole trend of
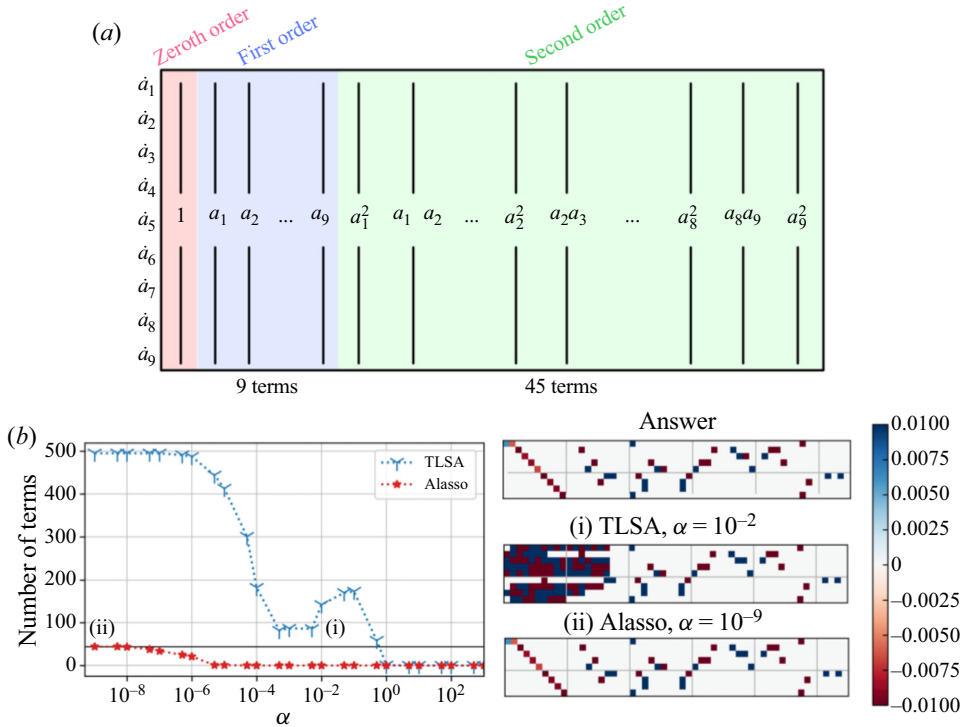
Figure 21. The SINDy for the nine-equation shear flow model. (a) Schematic of coefficient matrix β. (b) Relationship between the sparsity constant α and the number of terms with the obtained coefficient matrices.

Reference	SINDy
$\dot{a}_1 = 6.17 \times 10^{-3} - 6.17 \times 10^{-3} a_1 - 1.03 a_6 a_8 - 0.998 a_2 a_3$	$\dot{a}_1 = 6.15 \times 10^{-3} - 6.14 \times 10^{-3} a_1 - 1.03 a_6 a_8 - 0.995 a_2 a_3$
$\dot{a}_2 = 1.07 \times 10^{-2} a_2 - 1.03 a_1 a_3 - 1.03 a_3 a_9$	$\dot{a}_2 = 1.07 \times 10^{-2} a_2 - 1.03 a_1 a_3 - 1.03 a_3 a_9$
$\qquad + 1.22 a_4 a_6 - 0.365 a_5 a_7 - 0.149 a_5 a_8$	$\qquad + 1.21 a_4 a_6 - 0.365 a_5 a_7 - 0.149 a_5 a_8$
$\dot{a}_3 = -8.67 \times 10^{-3} a_3 + 0.308 a_4 a_7 - 0.390 a_4 a_8 + 0.308 a_5 a_6$	$\dot{a}_3 = -8.60 \times 10^{-3} a_3 + 0.308 a_4 a_7 - 5.56 \times 10^{-2} a_4 a_8 + 0.308 a_5 a_6$
$\dot{a}_4 = -8.85 \times 10^{-3} a_4 - 0.204 a_1 a_5 - 0.304 a_2 a_6$	$\dot{a}_4 = -8.76 \times 10^{-3} a_4 - 0.204 a_1 a_5 - 0.304 a_2 a_6$
$\qquad - 0.462 a_3 a_7 - 0.188 a_3 a_8 - 0.204 a_5 a_9$	$\qquad - 0.461 a_3 a_7 - 0.187 a_3 a_8 - 0.204 a_5 a_9$
$\dot{a}_5 = -6.79 \times 10^{-3} a_5 + 0.204 a_1 a_4 + 9.13 \times 10^{-2} a_2 a_7$	$\dot{a}_5 = -6.77 \times 10^{-3} a_5 + 0.204 a_1 a_4 + 9.08 \times 10^{-2} a_2 a_7$
$\qquad - 0.149 a_2 a_8 + 0.308 a_3 a_6 + 0.204 a_4 a_9$	$\qquad - 0.148 a_2 a_8 + 0.308 a_3 a_6 + 0.203 a_4 a_9$
$\dot{a}_6 = -1.13 \times 10^{-2} a_6 + 0.204 a_1 a_7 + 0.998 a_1 a_8 - 0.913 a_2 a_4$	$\dot{a}_6 = -1.13 \times 10^{-2} a_6 + 0.204 a_1 a_7 + 0.995 a_1 a_8 - 0.910 a_2 a_4$
$\qquad - 0.616 a_3 a_5 + 0.204 a_7 a_9 + 0.998 a_8 a_9$	$\qquad - 0.616 a_3 a_5 + 0.204 a_7 a_9 + 0.995 a_8 a_9$
$\dot{a}_7 = -9.29 \times 10^{-3} a_7 - 0.204 a_1 a_6 + 0.274 a_2 a_5$	$\dot{a}_7 = -9.27 \times 10^{-3} a_7 - 0.204 a_1 a_6 + 0.274 a_2 a_5$
$\qquad + 0.154 a_3 a_4 - 0.204 a_6 a_9$	$\qquad + 0.153 a_3 a_4 - 0.204 a_6 a_9$
$\dot{a}_8 = -9.29 \times 10^{-3} a_8 + 0.297 a_2 a_5 + 0.130 a_3 a_4$	$\dot{a}_8 = -9.23 \times 10^{-3} a_8 + 0.297 a_2 a_5 + 0.129 a_3 a_4$
$\dot{a}_9 = -5.55 \times 10^{-2} a_9 + 1.03 a_2 a_3 - 0.998 a_6 a_8$	$\dot{a}_9 = -5.54 \times 10^{-2} a_9 + 1.03 a_2 a_3 - 0.995 a_6 a_8$

Figure 22. Comparison of the governing equation for temporal coefficients.

coefficient matrix can be captured. These results imply that care should be taken for noisy observations in training data depending on the problem-setting users handling, although the SINDy can guarantee its robustness for a slight perturbation.
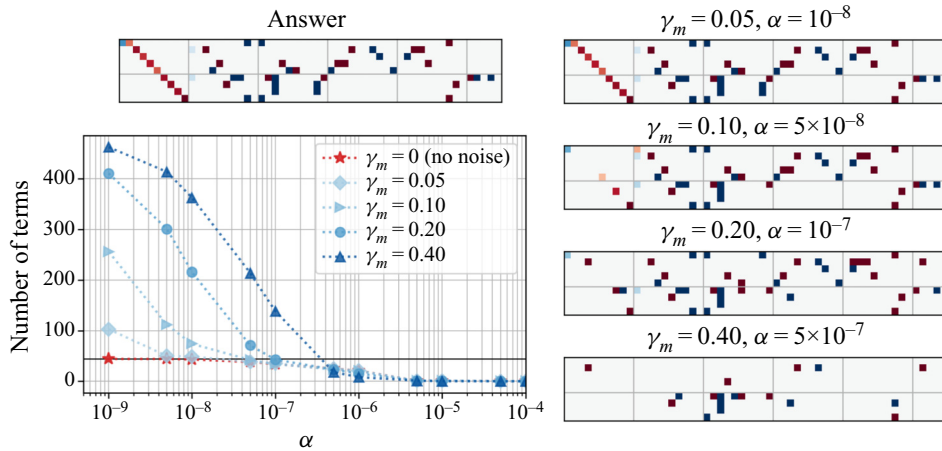
926 A10-22

Figure 23. Noise robustness of SINDy for the nine-shear turbulent flow example.

4. Conclusion

We performed a SINDy for low-dimensionalized fluid flows and investigated influences of the regression methods and parameter considered for construction of SINDy-based modelling. Following our preliminary test, the SINDys with two regression methods, the TLSA and the Alasso, were applied to the examples of a wake around a cylinder, its transient process, and a wake of two-parallel cylinders, with a CNN-AE. The CNN-AE was employed to map high-dimensional flow data into a two-dimensional latent space using nonlinear functions. Temporal evolution of the latent dynamics could be followed well by using SINDy with an appropriate parameter selection for both examples, although the required number of terms for the coefficient matrix are varied with each other due to the difference in complexity.

Finally, we also investigated the applicability of SINDy to turbulence with low-dimensional representation using a nine-shear flow model. The governing equation could be obtained successfully by utilizing Alasso with the appropriate sparsity constant. The results indicated that ML-ROM for turbulent flows can be perhaps presented in the interpretable sense in the form of an equation, if a well-designed model for mapping high-dimensional complex flows into low-dimensional manifolds can be constructed. With regard to the low-dimensional manifolds of fluid flows, the novel work by Loiseau *et al.* (2018*a*) discussed the nonlinear relationship between lower-order POD modes and higher-order counterparts of cylinder wake dynamics by relating it to a triadic interaction among POD modes. We may be able to borrow the concept here to investigate the possible nonlinear correlations in the data, although it will be tackled in the future since it is not easy to apply their idea directly to the nine-equation shear flow model which is composed of more complex relationships among modes. Otherwise, a combination of uncertainty quantification (Maulik *et al.* 2020), transfer learning (Guastoni *et al.* 2020*a*; Morimoto *et al.* 2020*b*) and Koopman-based frameworks (Eivazi *et al.* 2021) may also be possible extensions toward more practical applications.

The current results for CNN-SINDy modelling makes us sceptical about applications of CNN-AE to the flows where many spatial modes are required to represent the energetic field, e.g. turbulence, despite that recent several studies have shown its potential for unsteady laminar flows (Carlberg *et al.* 2019; Agostini 2020; Taira *et al.* 2020; Xu &

Duraisamy 2020; Bukka *et al.* 2021). Since a few studies have also recently recognized the challenges of the current form of CNN-AE for turbulence (Brunton, Hemati & Taira 2020; Fukami *et al.* 2020*c*; Glaws, King & Sprague 2020; Nakamura *et al.* 2021), care should be taken with the choice of low-dimensionalization tools depending on the flows that users handle. Hence, the utilization of other well-sophisticated low-dimensionalization tools such as spectral POD (Towne, Schmidt & Colonius 2018; Abreu *et al.* 2020), t-distributed stochastic neighbour embedding (Wu *et al.* 2017), hierarchical AE (Fukami *et al.* 2020*c*) and locally linear embedding (Roweis & Lawrence 2000; Ehlert *et al.* 2019) can be considered to achieve better mapping of flow fields. Taking constraints for coordinates extracted via AE is also a candidate to promote the combination of AE modes and conventional flow control tools (Champion *et al.* 2019*a*; Gelß *et al.* 2019; Ladjal *et al.* 2019; Jayaraman & Mamun 2020). Finally, we believe that the results of our examples, above, enables us to have a strong motivation for future work and notice the remarkable potential of SINDy for fluid dynamics.

Author ORCIDs.
ⓘ Kai Fukami https://orcid.org/0000-0002-1381-7322;
ⓘ Kai Zhang https://orcid.org/0000-0001-6097-7217;
ⓘ Koji Fukagata https://orcid.org/0000-0003-4805-238X.

Appendix A. Regression methods for SINDy

As introduced in § 1, the regression method used to obtain the coefficient matrix $\varXi$ is important since the resultant coefficient matrix $\varXi$ may vary with the choice of regression method. Here, let us consider a typical regression problem,

$$P = Q\beta, \tag{A1}$$

where P and Q denote the response variables and the explanatory variable, respectively. In this problem, we aim to obtain the coefficient matrix β representing the relationships between P and Q. Using the linear regression method, an optimized coefficient matrix β can be found by minimizing the error between the left-hand and right-hand sides such that

$$\beta = \operatorname{argmin} \|P - Q\beta\|^2. \tag{A2}$$

The TLSA, originally used in Brunton *et al.* (2016*a*) for performing SINDy, is based on this formulation in (A2) and obtains the sparse coefficient matrix by substituting zero for the coefficient below the threshold α. Its algorithm is summarized as algorithm 1:

Although the linear regression method has been widely used due to its simplicity, Hoerl & Kennard (1970) cautioned that estimated coefficients are sometimes large in absolute value for the non-orthogonal data. This is known as the overfitting problem. As a considerable surrogate method, they proposed the Ridge regression (Hoerl & Kennard 1970). In this method, the squared value of the components in the coefficient matrix is

---

Algorithm 1 Thresholded least square algorithm

1: $\alpha \leftarrow$ a certain value (set threshold)
2: **repeat**
3:     Obtain $\boldsymbol{\beta} = \text{argmin} \|\boldsymbol{P} - \boldsymbol{Q}\boldsymbol{\beta}\|^2$
4:     $\beta_i \leftarrow 0$ where $\beta_i < \alpha$
5:     Change $\boldsymbol{P}$ and $\boldsymbol{Q}$ so that the components which were set to zero in the previous step will be output as zero.
6: **until** the estimate $\boldsymbol{\beta}$ no longer changes.

---

considered as the constraint of the minimization process such that

$$\boldsymbol{\beta} = \text{argmin} \|\boldsymbol{P} - \boldsymbol{Q}\boldsymbol{\beta}\|^2 + \alpha \sum_j \beta_j^2, \tag{A3}$$

where α denotes the weight of the constraint term. For uses of SINDy, the coefficient matrix $\boldsymbol{\beta}$ is desired to be sparse, i.e. some components are zero, so as to avoid construction of a complex model and overfitting. As the sparse regression method, Lasso (Tibshirani 1996) was proposed. With Lasso, the sum of the absolute value of components in the coefficient matrix is incorporated to determine the coefficient matrix as follows:

$$\boldsymbol{\beta} = \text{argmin} \|\boldsymbol{P} - \boldsymbol{Q}\boldsymbol{\beta}\|^2 + \alpha \sum_j |\beta_j|. \tag{A4}$$

If the sparsity constant α is set to a high value, the estimation error becomes relatively large while the coefficient matrix results in a sparse matrix.

The Lasso algorithm introduced above has, however, two drawbacks and there are solutions for each: the elastic net (Enet) and the Alasso, as mentioned in the introduction. Enet (Zou & Hastie 2005) combines L_1 and L_2 errors, i.e. those of the Ridge regression, and Lasso. The optimal coefficient $\boldsymbol{\beta}$ is obtained as

$$\boldsymbol{\beta} = \left(1 + \frac{\alpha_2}{n}\right) \left\{ \text{argmin} \|\boldsymbol{P} - \boldsymbol{Q}\boldsymbol{\beta}\|^2 + \alpha_1 \sum_j |\beta_j| + \alpha_2 \sum_j \beta_j^2 \right\}. \tag{A5}$$

The following two parameters are set for the Enet: the sparsity constant $\alpha = \alpha_1 + \alpha_2$ and the sparse ratio $\alpha_1/(\alpha_1 + \alpha_2)$, respectively. The property for the sparsity constant α is the same as that of Lasso: a higher α results in large error and sparse estimation. The sparse ratio is the ratio of L_1 and L_2 errors as seen its definition. The sparse model can be obtained with a high L_1 ratio, although the collinearity problem may not be solved. Another solution, the Alasso (Zou 2006), uses adaptive weights for the penalizing coefficients in L_1 penalty term. In the Alasso, $\boldsymbol{\beta}$ are given by

$$\boldsymbol{\beta} = \text{argmin} \|\boldsymbol{P} - \boldsymbol{Q}\boldsymbol{\beta}\|^2 + \alpha \sum_j w_j |\beta_j|, \tag{A6}$$

where $w_j = (|\beta_j|)^{-\delta}$ denotes the adaptive weight with $\delta > 0$. The use of w_j enables us to avoid the issue of multiple local minima. The weights w_j and coefficients $\boldsymbol{\beta}$ are updated repeatedly like in algorithm 2:

In general it is necessary to choose the appropriate value for the threshold or the sparsity constant α in order to obtain a desired equation which is sparse and in an

---

Algorithm 2 Adaptive Lasso

1: $\alpha \leftarrow$ a certain value (set the sparsity constant)
2: $w_j \leftarrow 1$ (initialization)
3: **repeat**
4:     $Q^{**} \leftarrow Q/w_j$
5:     Solve the Lasso problem: $\beta^{**} = \text{argmin} \left\| P - Q^{**}\beta \right\|^2 + \alpha \sum_j w_j |\beta_j|$
6:     $\beta \leftarrow \beta^{**}/w_j$
7:     $w_j \leftarrow (|\beta_j|)^{-\delta}$
8: **until** the estimate β no longer changes.

---

interpretable form. Hence, we focus on seeking the appropriate α of each regression method in this study. The algorithm for the parameter search applied to the current study is described in algorithm 3. Regarding the assessment ways used in procedures 5 and 7

Algorithm 3 Parameter search

1: Prepare data X and $\dot{X}$
2: Prepare the set of sparse parameters α
3: **for all** α **do**
4:     Perform SINDy to obtain the ordinary differential equation at a certain α.
5:     Assess the candidate model with an evaluation index
6: **end for**
7: Choose the best model by the assessment
8: Model is identified.

---

in algorithm 3, number of total remaining terms of the obtained equation are utilized as the evaluation index for the preliminary test. For cylinder examples, the number of total terms, the error in the amplitudes of obtained waveforms, and the error ratio in the period between the obtained wave and the solution are assessed.

Appendix B. Preliminary tests with low-dimensional problems

B.1. *Pretest 1: van der Pol oscillator*

As a preliminary test of performing SINDy, we consider the van der Pol oscillator (Van der Pol 1934), which has a nonlinear damping. The governing equations are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = y, \tag{B1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -x + \kappa y - \kappa x^2 y, \tag{B2}$$

where $\kappa > 0$ is a constant parameter and we set $\kappa = 2$ in this preliminary test. The trajectory converges to a stable limit cycle determined by κ under any initial state except for the unstable point ($x = 0, y = 0$) as shown in figure 24.

Let us perform SINDy with each of the four regression methods, i.e. TLSA, Lasso, Enet and Alasso, with various sparsity constants α. Here, we use 20 000 discretized points as the training data sampled within $t = [0, 200]$ with the time step $\Delta t = 1 \times 10^{-2}$, although
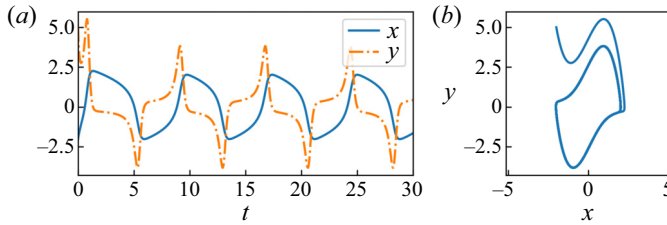
926 A10-26

Figure 24. Dynamics of the van der Pol oscillator with $\kappa = 2$: (*a*) time history and (*b*) trajectory on the *x*–*y* plane. The initial point is set to $(x, y) = (-2, 5)$ in this example.
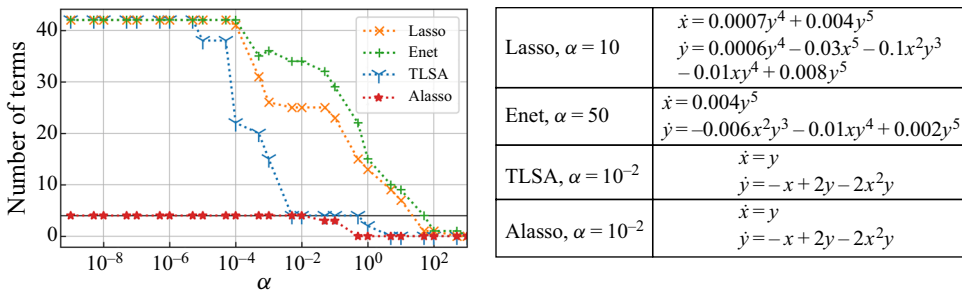

Lasso, $\alpha = 10$	$\dot{x} = 0.0007y^4 + 0.004y^5$ $\dot{y} = 0.0006y^4 - 0.03x^5 - 0.1x^2y^3$ $\quad - 0.01xy^4 + 0.008y^5$
Enet, $\alpha = 50$	$\dot{x} = 0.004y^5$ $\dot{y} = -0.006x^2y^3 - 0.01xy^4 + 0.002y^5$
TLSA, $\alpha = 10^{-2}$	$\dot{x} = y$ $\dot{y} = -x + 2y - 2x^2y$
Alasso, $\alpha = 10^{-2}$	$\dot{x} = y$ $\dot{y} = -x + 2y - 2x^2y$

Figure 25. Relationship between α and the number of terms, and the examples of obtained equations for four regression methods.

we will investigate the dependence of SINDy performance on both the length of time step and the length of training data range later. The library matrix is arranged including up to the fifth-order terms. In this example, we assess candidate models by using the total number of terms contained in the obtained ordinary differential equation. As denoted in (B1) and (B2), the true model has four non-zero terms in total in two governing equations. The relationship between the total number of terms in the candidate models and α is shown in figure 25. As shown, the uses of Lasso and Enet cannot reach the true model whose total number of terms is four. On the other hand, using TLSA or Alasso, the correct governing equation can be identified. It is striking that we should choose the appropriate parameter α for the correct model identification, as shown in figure 25. In this sense, the Alasso is suitable for the problem here since a wide range of α, i.e. $10^{-9} \leq \alpha \leq 10^{-2}$, can provide the correct equation. The difference here is likely due to the scheme of each regression. The TLSA is unable to discard the components of β with low α values: in other words, a lot of terms remain through the iterative process. On the other hand, the Alasso can identify the dominant terms clearly thanks to adaptive weights even if α is relatively low.

To investigate the dependence on the time step of the input data, SINDy is performed on the data with different time steps: $\Delta t = 1 \times 10^{-2}$, 0.1 and 0.2. Here, we consider only TLSA and Alasso following the aforementioned results. Note that the number of input data is different due to the time step since the integration time length is fixed to $t_{max} = 200$ in all cases. We have checked in our preliminary tests that this difference has no significant effect on the results. The trajectory, the results for the parameter search, and the example of obtained equations are summarized in figure 26. With the fine data, i.e. $\Delta t = 1 \times 10^{-2}$, the governing equations are correctly identified with some α for both methods. The true model can be found even if we use very small α, i.e. $\alpha = 1 \times 10^{-8}$ with Alasso as mentioned before. With increasing Δt, i.e. $\Delta t = 0.1$, the coefficients of terms are slightly underestimated. Additionally, the range of α where the number of terms is
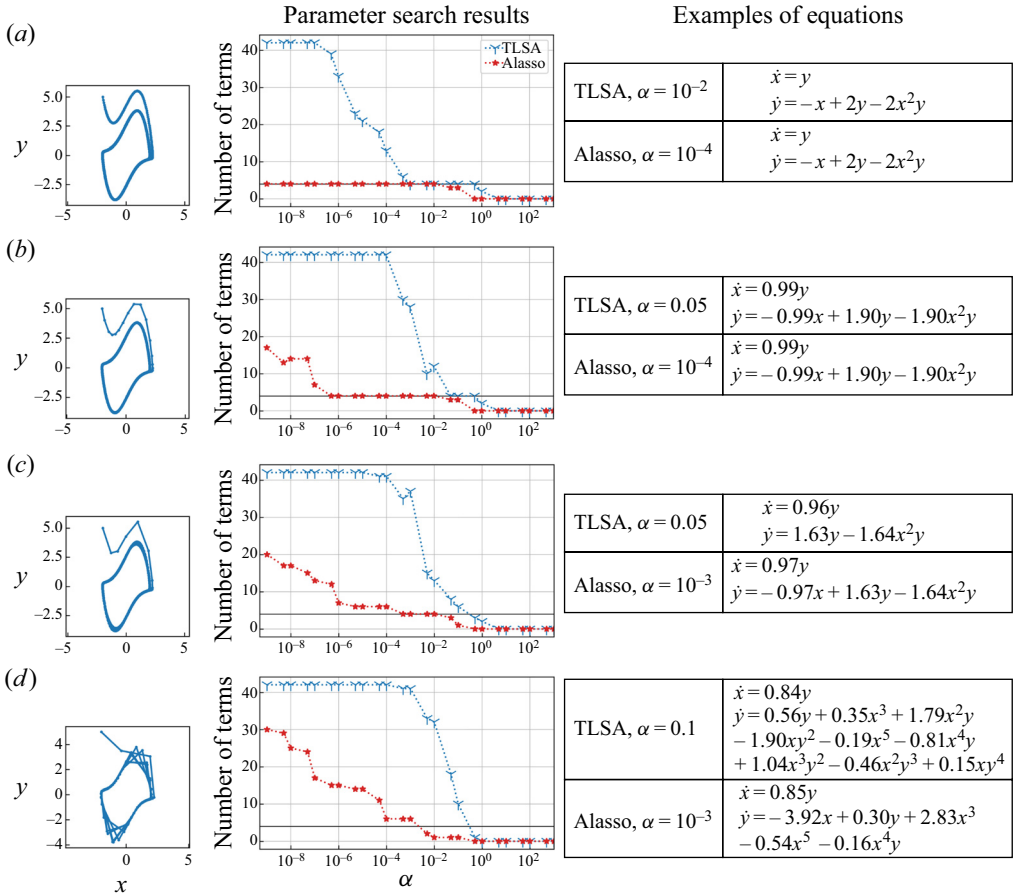
926 A10-27

Figure 26. Trajectory on the x–y plane, the relationship between α and number of terms, and the examples of obtained equations for data for different time steps: (*a*) $\Delta t = 1 \times 10^{-2}$, 769 points/period; (*b*) $\Delta t = 0.1$, 76.9 points/period; (*c*) $\Delta t = 0.2$, 38.5 points/period; (*d*) $\Delta t = 0.5$, 15.4 points/period.

correctly identified as four becomes narrower with both methods. With a wider time step, i.e. $\Delta t = 0.2$, the dominant terms can be found only with Alasso. On the other hand, the true model cannot be obtained with further wider time step data, i.e. $\Delta t = 0.5$, even with Alasso, due to the temporal coarseness as shown in figure 26(*d*). It suggests that the data with the appropriate time step is necessary for the construction of the model. In summary, Alasso is superior in terms of its ability to correctly identify the dominant terms for a wider range of sparsity constant.

Let us then investigate the dependence of the performance on the length of training data range, as demonstrated in figure 27. We here consider three lengths of training data range as (*a*) $t = [0, 8]$, (*b*) $t = [0, 12]$ and (*c*) $t = [0, 200]$ (baseline), while keeping the time step $\Delta t = 1 \times 10^{-2}$. As shown, the trends of the relationship between the sparsity constant and the number of terms in the identified equation are similar over the covered time length. In addition, the SINDys with the TLSA ($\alpha = 1 \times 10^{-2}$) and Alasso ($\alpha = 1 \times 10^{-3}$) are able to identify the equations successfully for all time length cases. Only in the case of $t = [0, 8]$, does the model identification not perform well with the Alasso
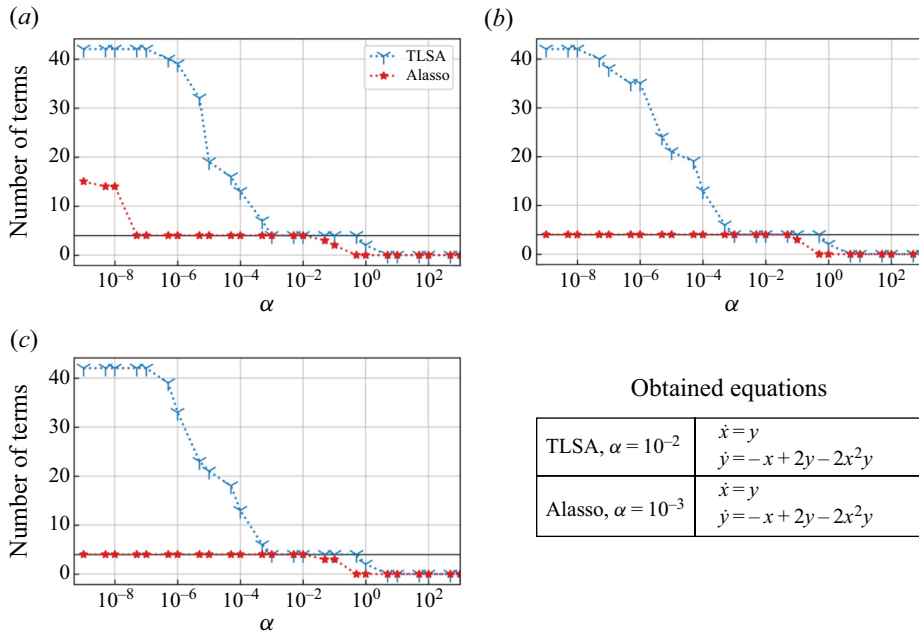
926 A10-28

Figure 27. The dependence of the performance of SINDy on the length of training data range: (*a*) $t = [0, 8]$; (*b*) $t = [0, 12]$; (*c*) $t = [0, 200]$ (baseline).

around $\alpha = 1 \times 10^{-8}$. The time length has little effect on the model identification if the time length is more than one period.

For considering models in which higher-order terms may be dominant due to higher nonlinearity, it will be valuable to find the way to select the appropriate potential terms that should be included in the library matrix. From this view, the dependence on the number of potential terms is investigated using the data with $\Delta t = 1 \times 10^{-2}$. In the case where the maximum order of the potential terms is the fifth, the true model can be found with two regression methods, i.e. TLSA and Alasso, as shown in figure 26(*a*). Here, the case with higher maximum order, i.e. the 15th and 20th, are also investigated as shown in figure 28. Note that there are 136 and 231 potential terms for each differential equation in these cases. For both cases with a different number of potential terms, we cannot find the true model with TLSA while the true model is identified in a wide range of α with Alasso. Furthermore, in the case including up to the 15th terms, some coefficients in the equation obtained by TLSA become huge. These large coefficients are due to the different functions included in the library being almost collinear.

B.2. *Pretest 2: Lorenz attractor*

As the second preliminary test, we consider the problem of identifying a chaotic trajectory called the Lorenz attractor (Lorenz 1963). It is written in a form of nonlinear ordinary differential equations,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\sigma x + \sigma y, \tag{B3}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \rho x - y - xz, \tag{B4}$$

(a)

TLSA, $\alpha = 10^{-2}$	$\dot{x} = y$ $\dot{y} = 1 \times 10^9 - x + 2y$ $-4 \times 10^{11} x^2 - 2x^2 y$
Alasso, $\alpha = 10^{-2}$	$\dot{x} = y$ $\dot{y} = -x + 2y - 2x^2 y$

(b)

TLSA, $\alpha = 0.1$	$\dot{x} = 0$ $\dot{y} = -0.1y - 0.4xy^2 + 0.2y^3 - 0.2x^2y^3$
Alasso, $\alpha = 10^{-4}$	$\dot{x} = y$ $\dot{y} = -x + 2y - 2x^2 y$
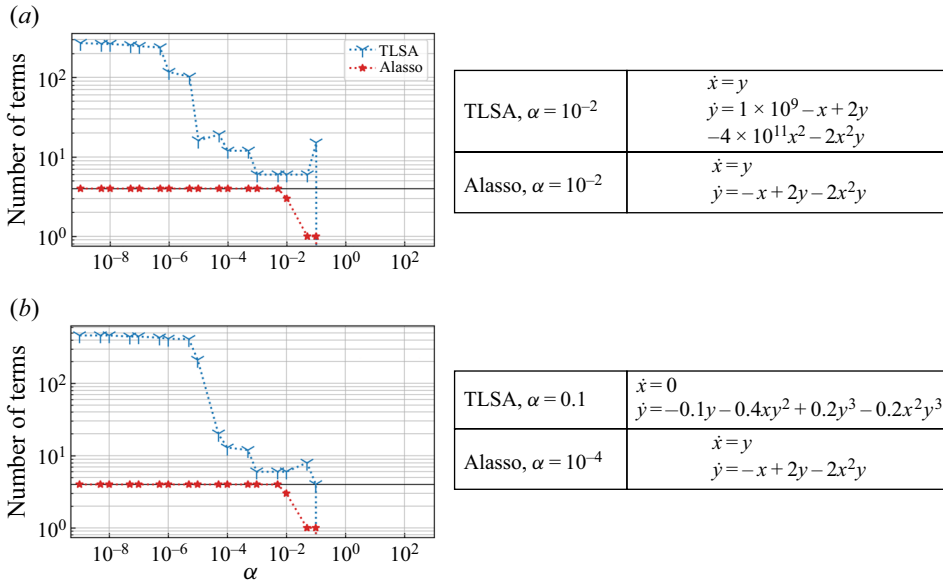
Figure 28. Relationship between α and number of terms and the examples of obtained equations for data for different library matrices: (a) including up to the 15th potential terms and (b) including up to the 20th potential terms.
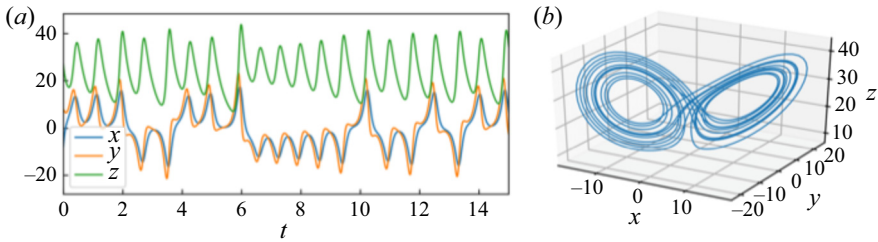
Figure 29. Dynamics of the Lorenz attractor: (a) time history and (b) trajectory.

$$\frac{\mathrm{d}z}{\mathrm{d}t} = -\iota z + xy. \tag{B5}$$

Lorenz (1963) showed that the dynamics governed by the equations above shows chaotic behaviour with the certain coefficients, i.e. $(\sigma, \rho, \iota) = (10, 28, 8/3)$ with the initial values $(x, y, z) = (-8, 8, 27)$, as shown in figure 29.

As with the preliminary test of the van der Pol oscillator, we consider four regression methods: TLSA, Lasso, Enet and Alasso, as shown in figure 30. Note that 10 000 discretized points with $\Delta t = 1 \times 10^{-2}$ from the time range of $t = [0, 100]$ are used as the training data and the potential terms up to fifth order are considered here. Similar to the van der Pol oscillator, we can obtain the true model whose total number of terms is seven with TLSA or Alasso, although Lasso or Enet cannot identify the equation correctly. Comparing the two methods which can obtain the correct model, the range of α, where the governing equations are identified, is wider using Alasso. We also perform SINDy with different time steps as shown in figure 31. Note that the considered time range is not changed depending on the time steps: in other words, the number of training data is changed per a considered time step, which is the same setting as that for the van der
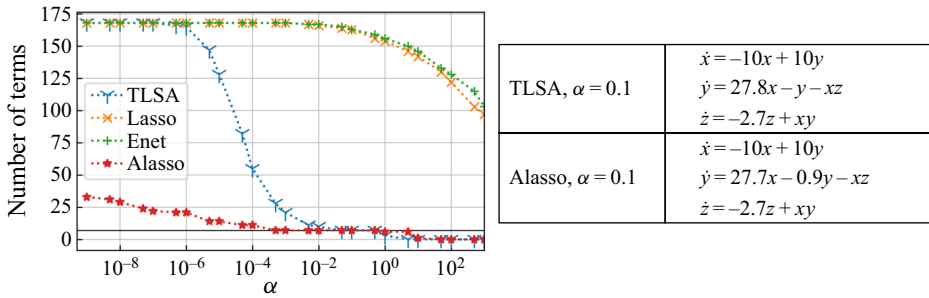
Figure 30. Relationship between α and the total number of terms, with the examples of obtained equations.
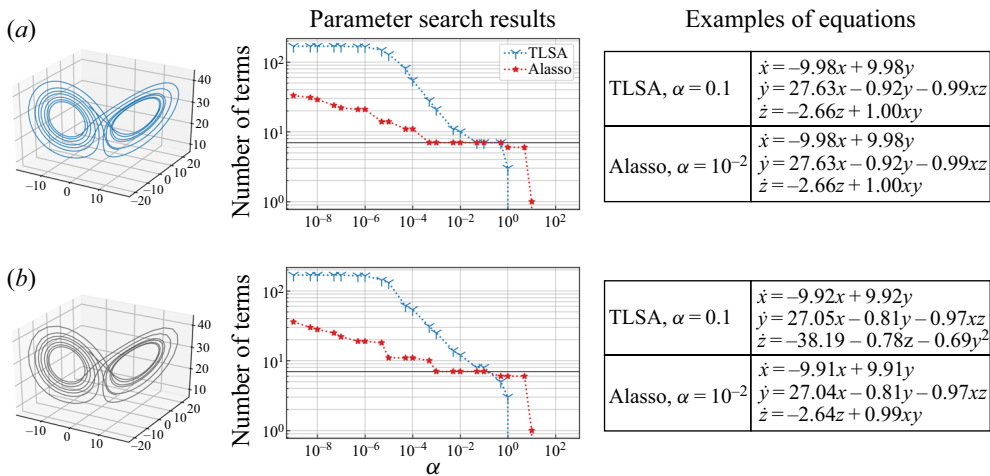

Figure 31. Trajectory on the x–y plane, the relationship between α and number of terms, and the examples of obtained equations for data for different time steps: (a) $\Delta t = 1 \times 10^{-2}$ and (b) $\Delta t = 2 \times 10^{-2}$.

Pol example. Similarly to the case of the van der Pol oscillator, we cannot reach the correct model using TLSA unless α is fine-tuned, and the coefficients are underestimated using Alasso with the wider time step data, i.e. $\Delta t = 2 \times 10^{-2}$. Moreover, the dependence on how many orders are considered in the library matrix is investigated in figure 32. There are 286 or 816 potential terms for each differential equation with the case including up to the 10th- or 15th-order terms, respectively. Analogous to the van der Pol oscillator example, the true model cannot be identified with TLSA; however, we can identify the true equation using Alasso even if there are as many as 816 potential terms with an appropriate range of α. Based on these results, TLSA and Alasso are selected as candidate regressions for high-dimensional complex flow problems.

In addition, we here note that the use of simple data processing (e.g. the standardization and the normalization) for both AE and SINDy pipelines may help Lasso and Enet to improve the regression ability, although it highly relies on a problem setting. Taking the data processing usually limits the model for practical uses because these data processing can only be successfully employed when we can guarantee an ergodicity, i.e. the statistical features of training data and test data are common with each other (Guastoni *et al.* 2020*b*; Morimoto *et al.* 2020*b*). When we perform the data processing, an inverse transformation error through the normalization or standardization is usually added since parameters for
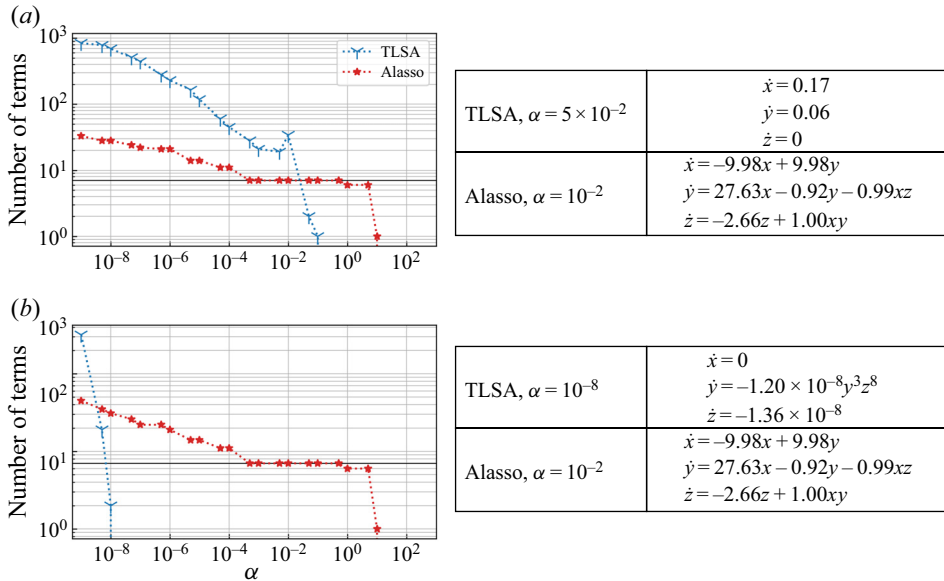
Figure 32. Relationship between α and number of terms, and the examples of obtained equations for data for different library matrices: (*a*) including up to the 10th potential terms and (*b*) including up to the 15th potential terms.

the processing, e.g. minimum value, maximum value, average value and variance, may be changed over the training and test phases. Hence, toward practical applications, it is better to prepare the regression method which is robust for the amplitudes of data oscillations.

REFERENCES

ABREU, L.I., CAVALIERI, A.V., SCHLATTER, P., VINUESA, R. & HENNINGSON, D.S. 2020 Spectral proper orthogonal decomposition and resolvent analysis of near-wall coherent structures in turbulent pipe flows. *J. Fluid Mech.* **900**, A11.

AGOSTINI, L. 2020 Exploration and prediction of fluid dynamical systems using auto-encoder technology. *Phys. Fluids* **32** (6), 067103.

BEETHAM, S. & CAPECELATRO, J. 2020 Formulating turbulence closures using sparse regression with embedded form invariance. *Phys. Rev. Fluids* **5** (8), 084611.

BEETHAM, S., FOX, R.O. & CAPECELATRO, J. 2021 Sparse identification of multiphase turbulence closures for coupled fluid–particle flows. *J. Fluid Mech.* **914**, A11.

BRUNTON, S.L., HEMATI, M.S. & TAIRA, K. 2020 Special issue on machine learning and data-driven methods in fluid dynamics. *Theor. Comput. Fluid Dyn.* **34**, 333–337.

BRUNTON, S.L. & KUTZ, J.N. 2019 *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2016*a* Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **113** (15), 3932–3937.

BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2016*b* Sparse identification of nonlinear dynamics with control. *IFAC NOLCOS* **49** (18), 710–715.

BUKKA, S.R., GUPTA, R., MAGEE, A.R. & JAIMAN, R.K. 2021 Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models. *Phys. Fluids* **33** (1), 013601.

CARLBERG, K.T., JAMESON, A., KOCHENDERFER, M.J., MORTON, J., PENG, L. & WITHERDEN, F.D. 2019 Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *J. Comput. Phys.* **395**, 105–124.

CHAMPION, K., LUSCH, B., KUTZ, J.N. & BRUNTON, S.L. 2019*a* Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116** (45), 22445–22451.

CHAMPION, K.P., BRUNTON, S.L. & KUTZ, J.N. 2019*b* Discovery of nonlinear multiscale systems: sampling strategies and embeddings. *SIAM J. Appl. Dyn. Syst.* **18** (1), 312–333.

DENG, N., NOACK, B.R., MORZYŃSKI, M. & PASTUR, L.R. 2020 Low-order model for successive bifurcations of the fluidic pinball. *J. Fluid Mech.* **884**, A37.

DURAISAMY, K. 2021 Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Phys. Rev. Fluids* **6**, 050504.

EHLERT, A., NAYERI, C.N., MORZYNSKI, M. & NOACK, B.R. 2019 Locally linear embedding for transient cylinder wakes. arXiv:1906.07822.

EIVAZI, H., GUASTONI, L., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2021 Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence. *Intl J. Heat Fluid Flow* **90**, 108816.

ENDO, K., TOMOBE, K. & YASUOKA, K. 2018 Multi-step time series generator for molecular dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, pp. 2192–2199. AAAI Press.

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019*a* Super-resolution analysis with machine learning for low-resolution flow data. In *11th International Symposium on Turbulence and Shear Flow Phenomena (TSFP11), Southampton, UK* (Paper 208).

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019*b* Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120.

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2020*a* Assessment of supervised machine learning for fluid flows. *Theor. Comput. Fluid Dyn.* **34** (4), 497–519.

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2021*a* Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* **909**, A9.

FUKAMI, K., HASEGAWA, K., NAKAMURA, T., MORIMOTO, M. & FUKAGATA, K. 2020*b* Model order reduction with neural networks: Application to laminar and turbulent flows. arXiv:2011.10277.

FUKAMI, K., MAULIK, R., RAMACHANDRA, N., FUKAGATA, K. & TAIRA, K. 2021*b* Global field reconstruction from sparse sensors with Voronoi tessellation-assisted deep learning. *Nat. Mach. Intell.* arXiv:2101.00554v1.

FUKAMI, K., NABAE, Y., KAWAI, K. & FUKAGATA, K. 2019*c* Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids* **4**, 064603.

FUKAMI, K., NAKAMURA, T. & FUKAGATA, K. 2020*c* Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Phys. Fluids* **32**, 095110.

GELSS, P., KLUS, S., EISERT, J. & SCHÜTTE, C. 2019 Multidimensional approximation of nonlinear dynamical systems. *J. Comput. Nonlinear Dyn.* **14** (6).

GLAWS, A., KING, R. & SPRAGUE, M. 2020 Deep learning for in situ data compression of large turbulent flow simulations. *Phys. Rev. Fluids* **5** (11), 114602.

GUASTONI, L., ENCINAR, M.P., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2020*a* Prediction of wall-bounded turbulence from wall quantities using convolutional neural networks. In *Journal of Physics: Conference Series*, vol. 1522, p. 012022. IOP Publishing.

GUASTONI, L., GÜEMES, A., IANIRO, A., DISCETTI, S., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2020*b* Convolutional-network models to predict wall-bounded turbulence from wall quantities. arXiv:2006.12483.

HASEGAWA, K., FUKAMI, K., MURATA, T. & FUKAGATA, K. 2019 Data-driven reduced order modeling of flows around two-dimensional bluff bodies of various shapes. In *ASME-JSME-KSME Joint Fluids Engineering Conference, San Francisco, USA* (Paper 5079).

HASEGAWA, K., FUKAMI, K., MURATA, T. & FUKAGATA, K. 2020*a* CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. *Fluid Dyn. Res.* **52** (6), 065501.

HASEGAWA, K., FUKAMI, K., MURATA, T. & FUKAGATA, K. 2020*b* Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theor. Comput. Fluid Dyn.* **34** (4), 367–388.

HINTON, G.E. & SALAKHUTDINOV, R.R. 2006 Reducing the dimensionality of data with neural networks. *Science* **313** (5786), 504–507.

HOERL, A.E. & KENNARD, R.W. 1970 Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12** (1), 55–67.

HOFFMANN, M., FRÖHNER, C. & NOÉ, F. 2019 Reactive SINDy: discovering governing reactions from concentration data. *J. Chem. Phys.* **150** (2), 025101.

JAYARAMAN, B. & MAMUN, S.M. 2020 On data-driven sparse sensing and linear estimation of fluid flows. *Sensors* **20** (13), 3752.

KAISER, E., KUTZ, J.N. & BRUNTON, S.L. 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc.* A **474** (2219), 20180335.

KANG, S. 2003 Characteristics of flow over two circular cylinders in a side-by-side arrangement at low Reynolds numbers. *Phys. Fluids* **15** (9), 2486–2498.

KINGMA, D.P. & BA, J. 2014 Adam: A method for stochastic optimization. arXiv:1412.6980.

KOR, H., BADRI GHOMIZAD, M. & FUKAGATA, K. 2017 A unified interpolation stencil for ghost-cell immersed boundary method for flow around complex geometries. *J. Fluid Sci. Technol.* **12** (1), JFST0011.

LADJAL, S., NEWSON, A. & PHAM, C. 2019 A PCA-like autoencoder. arXiv:1904.01277.

LECUN, Y., BOTTOU, L., BENGIO, Y. & HAFFNER, P. 1998 Gradient-based learning applied to document recognition. *Proc. IEEE* **86** (11), 2278–2324.

LI, S., KAISER, E., LAIMA, S., LI, H., BRUNTON, S.L. & KUTZ, J.N. 2019 Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems. *Phys. Rev.* E **100** (2), 022220.

LOISEAU, J.C. & BRUNTON, S.L. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67.

LOISEAU, J.-C., BRUNTON, S.L. & NOACK, B.R. 2018*a* From the POD-Galerkin method to sparse manifold models. https://doi.org/10.13140/RG.2.2.27965.31201.

LOISEAU, J.C., NOACK, B.R. & BRUNTON, S.L. 2018*b* Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490.

LORENZ, E.N. 1963 Deterministic nonperiodic flow. *J. Atmos. Sci.* **20** (2), 130–141.

LUMLEY, J.L. 1967 The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation* (ed. A.M. Yaglom & V.I. Tatarski). Nauka.

MATSUO, M., NAKAMURA, T., MORIMOTO, M., FUKAMI, K. & FUKAGATA, K. 2021 Supervised convolutional network for three-dimensional fluid data reconstruction from sectional flow fields with adaptive super-resolution assistance. arXiv:2103.09020.

MAULIK, R., FUKAMI, K., RAMACHANDRA, N., FUKAGATA, K. & TAIRA, K. 2020 Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids* **5**, 104401.

MILANO, M. & KOUMOUTSAKOS, P. 2002 Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26.

MOEHLIS, J., FAISST, H. & ECKHARDT, B. 2004 A low-dimensional model for turbulent shear flows. *New J. Phys.* **6** (56), 1–17.

MORIMOTO, M., FUKAMI, K. & FUKAGATA, K. 2020*a* Experimental velocity data estimation for imperfect particle images using machine learning. arXiv:2005.00756.

MORIMOTO, M., FUKAMI, K., ZHANG, K. & FUKAGATA, K. 2020*b* Generalization techniques of neural networks for fluid flow estimation. arXiv:2011.11911.

MORIMOTO, M., FUKAMI, K., ZHANG, K., NAIR, A.G. & FUKAGATA, K. 2021 Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low dimensionalization. *Theor. Comput. Fluid Dyn.* https://doi.org/10.1007/s00162-021-00580-0.

MURATA, T., FUKAMI, K. & FUKAGATA, K. 2020 Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **882**, A13.

NAKAMURA, T., FUKAMI, K., HASEGAWA, K., NABAE, Y. & FUKAGATA, K. 2021 Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* **33** (2), 025116.

OMATA, N. & SHIRAYAMA, S. 2019 A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder. *AIP Adv.* **9** (1), 015006.

ROWEIS, S. & LAWRENCE, S. 2000 Nonlinear dimensionality reduction by locally linear embedding. *Science* **290** (5500), 2323–2326.

RUDY, S.H., BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3** (4), e1602614.

SCHMELZER, M., DWIGHT, R.P. & CINNELLA, P. 2020 Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow Turbul. Combust.* **104** (2), 579–603.

SIPP, D. & LEBEDEV, A. 2007 Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows. *J. Fluid Mech.* **593**, 333–358.

SRINIVASAN, P.A., GUASTONI, L., AZIZPOUR, H., SCHLATTER, P. & VINUESA, R. 2019 Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **4**, 054603.

TAIRA, K., HEMATI, M.S., BRUNTON, S.L., SUN, Y., DURAISAMY, K., BAGHERI, S., DAWSON, S. & YEH, C.-A. 2020 Modal analysis of fluid flows: applications and outlook. *AIAA J.* **58** (3), 998–1022.

TIBSHIRANI, R. 1996 Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.* B **58** (1), 267–288.

TOWNE, A., SCHMIDT, O.T. & COLONIUS, T. 2018 Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J. Fluid Mech.* **847**, 821–867.

VAN DER POL, B. 1934 The nonlinear theory of electric oscillations. *Proc. Inst. Radio Engrs* **22** (9), 1051–1086.

WELLER, H.G., TABOR, G., JASAK, H. & FUREBY, C. 1998 A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **12** (6), 620–631.

WU, J., WANG, J., XIAO, H. & LING, J. 2017 Visualization of high dimensional turbulence simulation data using t-SNE. *AIAA Paper* 2017-1770.

XU, J. & DURAISAMY, K. 2020 Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. *Comput. Meth. Appl. Mech. Engng* **372**, 113379.

ZHANG, L. & SCHAEFFER, H. 2019 On the convergence of the SINDy algorithm. *Multiscale Model. Simul.* **17** (3), 948–972.

ZOU, H. 2006 The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* **101** (476), 1418–1429.

ZOU, H. & HASTIE, T. 2005 Regularization and variable selection via the elastic net. *J. R. Stat. Soc.* B **67** (2), 301–320.