Nonlinear model reduction by deep autoencoder of noise response data

Kenji Kashima[1,2]

***Abstract*— In this paper a novel model order reduction method for nonlinear systems is proposed. Differently from existing ones, the proposed method provides a suitable *nonlinear* projection, which we refer to as *control-oriented deep autoencoder (CoDA)*, in an easily implementable manner. This is done by combining noise response data based model reduction, whose control theoretic optimality was recently proven by the author, with stacked autoencoder design via deep learning.**

I. INTRODUCTION

Nowadays, various mathematical models for dynamical behavior of complex systems are available, e.g., biological systems, smart grids, social networks [1]. For these models, standard analysis and design methods are not directly applicable. The major reason for this restriction is their high dimensionality and nonlinearity. In order to manage such situations, nonlinear model reduction is envisaged to be a powerful tool. However, although research of nonlinear model reduction has a long history, its applicability is still limited [2], [3].

What we focus on this paper is "model reduction of nonlinear systems *by nonlinear state transform*". First, when the dynamics are linear the use of linear transforms is often theoretically justifiable and there are many well established methods [4]. Concerning nonlinear systems, control theoretic model reduction methods require an infeasible computational burden, e.g., to solve nonlinear partial differential equations [5], [6]. As a result, this approach has not yet lead to the development of practical tools. On the other hand, simulation-based approaches such as Proper Orthogonal Decomposition (POD), have been widely utilized in computational physics and numerical analysis [7], [8]. Although this is applicable to complex systems such as nonlinear partial differential equations, the approximation accuracy, in the sense of the input-output system, is not taken into explicit account. In addition, this is model reduction by linear transforms.

In view of this, we attempt to develop a practical method to reduce nonlinear systems by possibly nonlinear transforms. To tackle this challenging problem, we utilize two important building blocks: One is the control theoretic optimality of the simulation-based model reduction method [9], [10]. This result can provide a simulation-based approach with an approximation gap evaluation in the sense of input-output systems. Although only linear transforms are considered in [9], [10], it can be extended to nonlinear ones; see Theorem 1 below. The other is deep learning, which has been utilized in many fields in recent years [11], [12]. We reformulate model reduction as a learning problem, for which a stacked autoencoder can provide a practical way to find a reasonable nonlinear state transform.

This paper is organized as follows: In Section II we introduce the fundamental idea of model reduction and its simulation-based method. In Section III we compare controllability-based and simulation-based model reduction methods. In Section IV we reformulate the model reduction problem as a learning problem, which leads to a novel model reduction method by possibly nonlinear projections. In Section V we examine the usefulness of the proposed method through a numerical example. Some concluding remarks are given in Section VI.

Notation: The set of real numbers is $\mathbb{R}$. For a vector x, the Euclidean norm is $\|x\|$. For a matrix X, the transpose is X^{T}. Let w_t be the standard Wiener process (of suitable dimension) which has a unit covariance matrix [13]. The associated filtration, probability and expectation are denoted by $\mathcal{F}_t$, $\mathbb{P}$ and $\mathbb{E}$, respectively.

II. PRELIMINARIES

In this section we briefly review some basic concepts of projection-based model reduction.

A. Galerkin projection

Suppose that the system to be reduced is given as

$$\frac{d}{dt}x(t) = f(x(t)) + g(x(t))u(t),\ x(0) = x_{\rm i}. \qquad (1)$$

We attempt to reduce the dimension n of the state variable $x(t)$ to $k(< n)$. Assume that the smooth mappings $\varrho : \mathbb{R}^n \to \mathbb{R}^k$ and $\varrho^\dagger : \mathbb{R}^k \to \mathbb{R}^n$ satisfy

$$x(t) - (\varrho^\dagger \circ \varrho)(x(t)) \approx 0,\ {}^\forall t \qquad (2)$$

for the solution of (1), where $\circ$ represents the composite mapping. Then, the reduced state variable $z(t) := \varrho(x(t)) \in \mathbb{R}^k$ can recover the original one by $x(t) \approx \varrho^\dagger(z(t)) =: \hat{x}(t)$ approximately. In this sense, the dynamics of k-dimensional variable $z(t)$ can be considered to be a suitable reduced model of (1).

In particular, when the mappings are linear, i.e., ϱ and $\varrho^\dagger$ are matrices, the dynamics of $z(t)$ can be written as

$$\frac{d}{dt}z(t) = \varrho f(\varrho^\dagger z(t)) + \varrho g(\varrho^\dagger z(t))u(t),\ z(0) = \varrho x_{\rm i}. \quad (3)$$

This form of linear projection $\varrho^\dagger\varrho \in \mathbb{R}^{n\times n}$ is called a Petrov-Galerkin projection and covers most of the standard model reduction methods [4].

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
[2] Japan Science and Technology Agency, CREST 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan
kashima@amp.i.kyoto-u.ac.jp

We can say that the goal of model reduction is to find ϱ and $\varrho^\dagger$ that satisfy (2) in a suitable sense. In the control theoretic approach we attempt to achieve (2) for trajectories that can be realized by low energy control inputs. Controllability analysis plays an important role for this purpose. However, the practical applicability in this line of methods is still limited to linear projections for linear systems [4].

B. Proper Orthogonal Decomposition

It also seems reasonable to find matrices ϱ and $\varrho^\dagger$ that minimize

$$\sum_{\tau\in\mathcal{T}} \|x(\tau) - \varrho^\dagger\varrho x(\tau)\|^2 \tag{4}$$

where the error is evaluated at multiple, given time instances $\tau \in \mathcal{T}$. This is the fundamental idea of the POD, and is referred to as the *method of snapshots*.

Note that this optimization only requires numerical simulation of the dynamics, and least square error analysis. That is, define $X := [x(t_1)\ x(t_2)\ \cdots\ x(t_N)] \in \mathbb{R}^{n\times N}$ where $\mathcal{T} = \{t_1, \cdots, t_N\}$, and apply its singular value decomposition to obtain $X = U\Sigma V^\top$. Then, the optimal k-dimensional ϱ is given as $[u_1, \ldots, u_k]$ where u_i is the left-singular vector corresponding to the i-th largest singular value. Note that this procedure is computationally tractable even for nonlinear large-scale systems [7]. However, it is generally impossible to simulate the trajectories for all admissible input signals. Thus, we need to carefully choose input signals when we collect snapshots.

III. Control theoretic optimality of the simulation-based method

A. Advantages and disadvantages

Let us summarize the advantages and disadvantages of the controllability-based and simulation-based model reduction methods from the following aspects:

1) Approximation as input-output systems: In the control theoretic approach the approximation error is small at states that are reachable with small control effort. On the other hand, in the simulation-based approach a similar evaluation is available for linear dynamics only by collecting snapshots, achieved by injecting impulse or sinusoidal inputs [14], [15]. However, there is no theoretic guarantee for nonlinear dynamics.
2) Applicability to nonlinear dynamics: The simulation-based method does not require linearity of the dynamics as far as its simulation is easily executable. However, for the controllability-based reduction of nonlinear dynamics one needs to solve a nonlinear partial differential equation with high-dimensional spatial variables, which is not generally feasible.
3) Nonlinear projection: In both cases there is no practical method to search for nonlinear projections.

We consider 1) and 2) in Section III-B, and 3) in Section IV.

B. Control theoretic evaluation for simulation-based model reduction

In this section we confirm that the noise response data can provide both a practical computational method and also control theoretic evaluation. In the case of the POD, let us utilize the noise response data, that is, the snapshots of (1) with white noise as its input:

$$\begin{aligned} d\bar{x}_t &= f(\bar{x}_t)dt + g(\bar{x}_t)dw_t, \\ \bar{x}_0 &= x_{\rm i}. \end{aligned} \tag{5}$$

Then, we attempt to minimize

$$J_{\rm POD}(\varrho, \varrho^\dagger) := \sum_{\tau\in\mathcal{T}} \mathbb{E}\left[\|(I - \varrho^\dagger\circ\varrho)\bar{x}_\tau\|^2\right]. \tag{6}$$

Throughout this paper, we assume that (5) has a unique solution process in a suitable sense.

Next, concerning the controllability-based approach, let us investigate the following controlled dynamics, whose input is disturbed by random noise:

$$\begin{aligned} dx_t &= f(x_t)dt + g(x_t)(u_t dt + dw_t), \\ x_0 &= x_{\rm i}. \end{aligned} \tag{7}$$

In order to quantify the reachability of a state $\tilde{x} \in \mathbb{R}^n$ at time τ, it is standard to investigate the minimum input energy to achieve $x(\tau) = \tilde{x}$, for both linear and nonlinear systems [4], [6]. Such a minimum input energy (of $\tilde{x}$) is called a controllability function. The author introduced a definition of a controllability function that can incorporate the effects of stochasticity:

Definition 1 ([10]): Given (7) and $\gamma > 0$, define the γ-stochastic controllability function as

$$L_\gamma(\tau, \tilde{x}) := \inf_{u\in\mathcal{U}} \mathbb{E}\left[\int_0^\tau \frac{1}{2}\|u_t\|^2 dt + \frac{\gamma}{2}\|x_\tau - \tilde{x}\|^2\right], \tag{8}$$

where $u \in \mathcal{U}$ means u_t is $\mathcal{F}_t$-measurable for all $t \in [0, \tau]$. ■

Here, the infimum is taken over all *causal feedback control laws*. Note that the terminal constraint $x_\tau = \tilde{x}$ was replaced by a quadratic cost. For sufficiently large γ, $L_\gamma(\tau, \tilde{x})$ characterizes the difficulty to traverse from $x_{\rm i}$ to a neighborhood of $\tilde{x}$ at time τ in terms of the required control energy under the existence of noise. Actually, in the large γ limit, the terminal cost assigns $+\infty$ for $x_\tau \neq \tilde{x}$ and 0 for $x_\tau = \tilde{x}$. In this sense, the allowable neighborhood shrinks to the point $\tilde{x}$, which plays the role of the fixed terminal condition.

For later discussion, we combine this measure with Definition 1 to define the best projection to reduce the dimension of stochastic system (7):

Definition 2: Given $L_\gamma(\tau, \tilde{x})$, we define

$$J(\varrho, \varrho^\dagger) := \sum_{\tau\in\mathcal{T}} \lim_{\gamma\to\infty} \int_{\mathbb{R}^n} \frac{{\rm e}^{-L_\gamma(\tau,\tilde{x})}}{\int_{\mathbb{R}^n} {\rm e}^{-L_\gamma(\tau,\tilde{x})} d\tilde{x}} \|(I - \varrho^\dagger\circ\varrho)\tilde{x}\|^2 d\tilde{x}. \tag{9}$$

■

Minimization of $J(\varrho, \varrho^\dagger)$ yields a projection that realizes a smaller projection error at the every state $\tilde{x}$ that is reachable at time τ by a small input energy (i.e., $L_\gamma(\tau, \tilde{x})$ is small).

We are now ready to explain the first building block:

Theorem 1: Under the definitions above,

$$J(\varrho, \varrho^\dagger) = J_{\rm POD}(\varrho, \varrho^\dagger) \tag{10}$$

for any smooth mapping ϱ and $\varrho^\dagger$. ■

Proof: This equality was shown in [10] by assuming ϱ and $\varrho^\dagger$ are linear. The proof is valid for nonlinear mappings if they are smooth. The details are omitted. ■

Therefore, the noise response data fitting in (6) is equivalent to the minimization of the control theoretic error criterion (9).

IV. Reduction by autoencoder

In what follows, we attempt to find possibly nonlinear mappings that minimize (6). A natural way to do this may be fixing some basis functions, and then to optimize parameters therein. However, the choice of basis functions is highly nontrivial. In this paper, we propose to utilize techniques developed in learning theory. This does not require a priori specification of the nonlinearity.

A. Reformulation as a learning problem

It is convenient for later discussion to describe the minimization of (6) as a learning problem. Let us denote the noise response data set

$$\mathcal{X}_S := \{\bar{x}_t^{(s)} : s = 1, 2, \ldots, S,\ t \in \mathcal{T}\} \tag{11}$$

where s and S denote the label and total number of sample paths. Define

$$J_{\rm DL}^S(\varrho, \varrho^\dagger) := \frac{1}{S} \sum_{x \in \mathcal{X}_S} \|x - (\varrho^\dagger \circ \varrho)x\|^2. \tag{12}$$

Theorem 2: Under the notation above,

$$\lim_{S \to +\infty} J_{\rm DL}^S(\varrho, \varrho^\dagger) = J(\varrho, \varrho^\dagger) \tag{13}$$

for arbitrary ϱ and $\varrho^\dagger$. ■

Proof: The result is obtained by the central limit theorem and Theorem 1. ■

The minimization of (12) suggests a link to learning theory, which will be briefly reviewed in the next section.

B. Autoencoder

In this section, we briefly explain learning by neural networks, in particular an autoencoder. Let us consider an artificial neuronal component as shown in Fig. 1. This unit receives input signals $x_1,\ x_2,\ \ldots,\ x_n$ and produces the output z. The actual mapping is given by

$$z = a(\boldsymbol{w}^\top x + b) = a\bigg(\sum_{i=1}^n w_i x_i + b\bigg)$$

where constant vector $\boldsymbol{w} = [w_1\ \cdots\ w_n]^\top$ contains weighting parameters, b is bias, and the nonlinear function a is called an activation function. In this paper a layered interconnection of such neurons is referred to as a neural network.

Next, let us consider the simple neural network shown in Fig. 2. Note that the number of neurons of the input (left) layer and the output (right) layer are the same, and that the middle layer has fewer neurons. The neurons in the input and output layers are identical mappings. In addition, we put together the weight and bias between input-middle and middle-output layers as

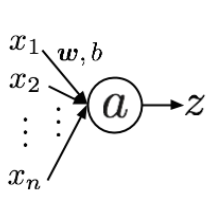

Fig. 1. Artificial neuron

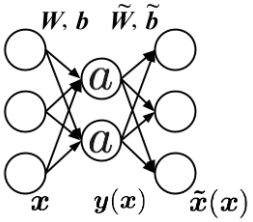

Fig. 2. Autoencoder

$$\boldsymbol{W} = [\boldsymbol{w}_1\ \cdots\ \boldsymbol{w}_k]^\top \in \mathbb{R}^{k \times n} \tag{14}$$

$$\boldsymbol{b} = [b_1 \cdots b_k]^\top \in \mathbb{R}^k \tag{15}$$

$$\tilde{\boldsymbol{W}} = [\tilde{\boldsymbol{w}}_1\ \cdots\ \tilde{\boldsymbol{w}}_n]^\top \in \mathbb{R}^{n \times k} \tag{16}$$

$$\tilde{\boldsymbol{b}} = [\tilde{b}_1\ \cdots\ \tilde{b}_n]^\top \in \mathbb{R}^n. \tag{17}$$

Then, the output $\boldsymbol{y}(x) \in \mathbb{R}^k$ of the middle and right-most layers is given by

$$\boldsymbol{y}(x) := \boldsymbol{a}(\boldsymbol{W}x + \boldsymbol{b}),$$

and

$$\begin{aligned} \tilde{\boldsymbol{x}}(x) &:= \tilde{\boldsymbol{W}}\boldsymbol{y}(x) + \tilde{\boldsymbol{b}} \\ &= \tilde{\boldsymbol{W}}\boldsymbol{a}(\boldsymbol{W}x + \boldsymbol{b}) + \tilde{\boldsymbol{b}}, \end{aligned}$$

where $\boldsymbol{a}(\cdot)$ represents the nonlinear function acting on $\mathbb{R}^k$ consisting of the activation function a.

A neural network whose output signals approximate the input signal well are called autoencoder. In the simple neural network above, the learning as an autoencoder for given data set $\mathcal{X}$ can be regarded as the following minimization problem:

$$\min_{\boldsymbol{W}, \boldsymbol{b}, \tilde{\boldsymbol{W}}, \tilde{\boldsymbol{b}}} \sum_{x \in \mathcal{X}} \|x - \tilde{\boldsymbol{x}}(x)\|^2. \tag{18}$$

This can be viewed as a nonlinear principal component analysis of the underlying probability distribution of the given data set. An important feature is that we do not need to care much about the nonlinearity, i.e., the choice of the activation function. Actually, the standard choice for this is the sigmoid function $a(x) = 1/(1 + \mathrm{e}^{-x})$, or the rectifier $a(x) = \max(x, 0)$ independent of the given data set. Instead, we expect that suitable nonlinear mappings that make (18) small are obtained somewhat automatically, assuming a large number of neurons and the availability of a large amount of training data.

C. Nonlinear projection as a deep autoencoder

Now, we characterize a relation between model reduction problem and autoencoder learning. Actually, the noise response data fitting in Section III is in the same form as (18). Thus, we propose to minimize (18) where

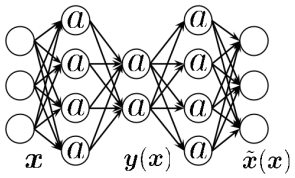

Fig. 3. Deep autoencoder

- the training data set is the noise response data $\mathcal{X}_S$ in (11) for large S, and
- the number of neurons of the middle layer is the desired dimension k of the reduced model.

Then, once an autoencoder is successfully achieved, we can use

- the mapping from the input signal to the middle layer as ϱ, and
- the mapping from the middle layer to the output as $\varrho^\dagger$,

respectively, that is,

$$\varrho^\dagger(x) := \boldsymbol{a}(\boldsymbol{W}x + \boldsymbol{b}) = \boldsymbol{y}(x) \tag{19}$$
$$\varrho(z) := \tilde{\boldsymbol{W}}z + \tilde{\boldsymbol{b}} = \tilde{\boldsymbol{x}}(x). \tag{20}$$

In this paper, in view of Theorem 2, the autoencoder trained by the noise response data is referred to as *Control-oriented Autoencoder* (abbr. **CoA**).

In practice we utilize an autoencoder having multiple layers. This is called a *deep (or stacked) autoencoder*, and has played a crucial role in the recent application of deep learning. Learning of such multi-layer neural networks had been known to be difficult because back propagation, the most standard procedure to update parameters, does not work properly. To circumvent this issue, layer-wise learning with the help of autoencoders was a breakthrough [16]. This successfully made deep learning an outstanding feature extraction method in various fields [11], [12], [17]. Hereafter, **CoA** with multi-layer structure is referred to as *Control-oriented Deep Autoencoder* (abbr. **CoDA**). We expect that highly nontrivial projections, which cannot, realistically, be characterized by an analytical method, can be obtained also in the **CoDA**.

It should be noted that the use of noisy training data is known to contribute to better learning performance, i.e., the result is robustified by avoiding overfitting [11], [12]. In the proposed method, the training data is the snapshot of the noise response. Therefore, we can expect that the resulting projection has a similar enhanced generalization ability to input signals.

V. Numerical Example

A. Reduction of a singularly perturbed system

We examine the potential usefulness of the proposed **CoDA** through a numerical example. As stated in Section I, there is no general method for model reduction by nonlinear state transforms. For comparison, we investigate the following nonlinear system:

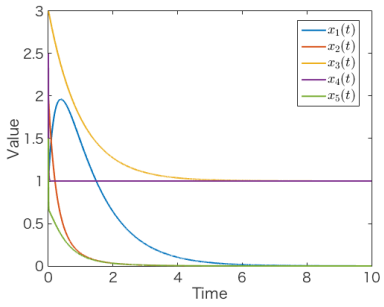

Fig. 4. Time response of the original model

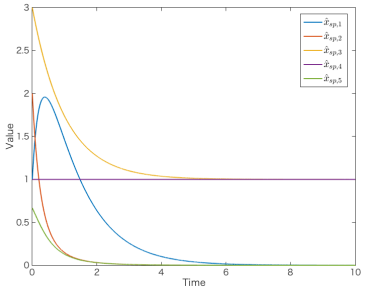

Fig. 5. Time response of the singular perturbation model

$$\begin{cases} \dot{x}_1 = -x_1 + x_2 + x_2x_3 \\ \dot{x}_2 = -x_2 - x_2x_3 + x_5 \\ \dot{x}_3 = -x_3 + x_4 + u \\ \dot{x}_4 = \frac{1}{\varepsilon}(x_4 - x_4^2) \\ \dot{x}_5 = \frac{1}{\varepsilon}(x_2 - x_5 - x_2x_5) \end{cases} \tag{21}$$

with $x_{\rm i} = [1\ 2\ 3\ 2.5\ 1.5]^\top$ where each element of the state variable $x \in \mathbb{R}^5$ is positive, ε is a sufficiently small positive constant, and u is the exogenous input. This is taken from a model of chemical reactions in a living organism. The time response with $\varepsilon = 0.01$ and $u = 0$ is shown in Fig. 4.

We can apply singular perturbation to the reduction of this nonlinear system since the time scale are explicitly separated [18]. That is, by solving $\dot{x}_4 = 0,\ \dot{x}_5 = 0$, we obtain

$$\begin{aligned} \varrho^\dagger_{\rm sp}(x) &:= [x_1\ x_2\ x_3]^\top \\ \varrho_{\rm sp}(z) &:= [z_1\ z_2\ z_3\ 1\ \frac{z_2}{1+z_2}]^\top. \end{aligned}$$

This ignores the fast dynamics, but approximates slow dynamics well; see Fig. 5. We use this nonlinear mapping for the comparison.

For the **CoDA**, the noise response snapshots (11) with

$$S = 200,\ \mathcal{T} = \{0.01 \times (k-1) : k = 1, 2, \ldots, 1000\}$$

are collected for the training data. The total number of data is 2.0×10^5. We utilized the aforementioned layer-wise learning via back propagation to achieve a deep autoencoder.

TABLE I

COMPARISON WITH POD

	$J^S_{\mathrm{DL}}(\varrho, \varrho^\dagger)$
CoDA ($k = 2$)	0.22875
POD	71.6

TABLE II

COMPARISON WITH SINGULAR PERTURBATION

	$J^S_{\mathrm{DL}}(\varrho, \varrho^\dagger)$
CoDA ($k = 3$)	0.01376
Singular perturbation	3.009

B. Comparison with POD

In this section we set $k = 2$ and compare the result with POD, i.e., linear projections. Table I shows the projection error $J^S_{\mathrm{DL}}(\varrho, \varrho^\dagger)$ in (12) for the **CoDA** and POD. This value for the POD is the minimum achievable one among all the linear mappings. Therefore, this result implies the existence of a 2-dimensional *nonlinear manifold* that approximates the noise response data to a high level of accuracy that cannot be realized by any 2-dimensional linear subspace. This shows the potential for model reduction by nonlinear mappings to outperform the POD.

C. Comparison with singular perturbation

Next, we compare with the singular perturbation to observe how the proposed approach achieves a reasonable nonlinear mapping without using a priori information of the explicit time scale separation, that is, only from the noise response data. For this purpose, we fix $k = 3$ and the number of neurons in each layer is given by

$$\mathbf{5} - 15 - 7 - \mathbf{3} - 7 - 15 - \mathbf{5}.$$

First, we impose some structure on ϱ and $\varrho^\dagger$ in order to examine the shape of the obtained nonlinear mapping. That is, let $\varrho = \varrho_{\mathrm{sp}}$, $\varrho^\dagger = \varrho^\dagger_{\mathrm{sp}}$ except for the fifth entry of $\varrho^\dagger$. The fifth entry, denoted by $\varrho^\dagger_5(z_2)$, is restricted to a function of z_2, and is to be designed via deep autoencoder design. Table II shows the projection error $J^S_{\mathrm{DL}}(\varrho, \varrho^\dagger)$ in (12). We can observe that the **CoDA** achieves a better fitting performance without a priori information on the time scale separation property. Fig. 6 shows

$$\varrho^\dagger_5(z_2), \ \frac{z_2}{1+z_2}, \ \text{and} \ (x_2(t), x_5(t)) \ \text{for} \ \mathcal{X}_S.$$

We can observe that the obtained nonlinear projection is close to the result of the singular perturbation. This implies that the **CoDA** can extract the slow dynamics that are well captured by the singular perturbation. However, there is a large mismatch between them around $1.8 \le z_2 \le 2$. This region corresponds to the transient response just after the initial time, where the fast dynamics are still dominant; see Fig. 4. In the case of the singular perturbation, such dynamics are ignored. On the other hand, the **CoDA** shows a good accordance during these fast dynamics.

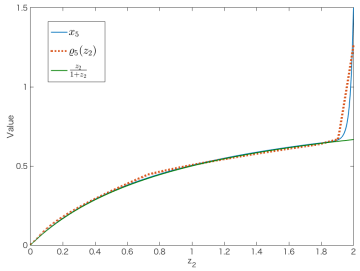

Fig. 6. Comparison between $\varrho_5(z_2)$, $\frac{z_2}{1+z_2}$, and $(x_2(t), x_5(t))$ for $\mathcal{X}_S$.

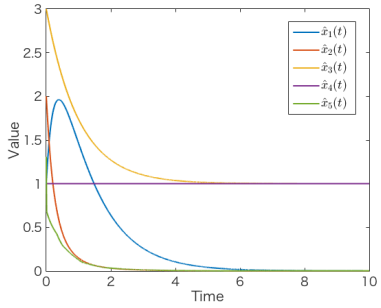

Fig. 7. Time response of the reduced order model obtained by **CoDA**

Next, the constraints on $\varrho^\dagger_4$ and $\varrho^\dagger_5$ are removed. We examine dynamical state trajectories of the original and reduced systems. As a nonlinear counterpart of the Petrov-Galerkin projection, the reduced dynamics are given by the time differentiation of $z(t) := \varrho^\dagger(x(t))$ such that

$$\begin{aligned} \dot{z}(t) &= \frac{\partial \varrho}{\partial x} \cdot \frac{dx}{dt} \\ &\approx \left.\frac{\partial \varrho}{\partial x}\right|_{x=\varrho(z(t))} \cdot \big(f(\varrho^\dagger(z(t))) + g(\varrho^\dagger(z(t))u(t))\big) \\ \hat{x}(t) &= \varrho^\dagger(z(t)). \end{aligned}$$

The time discretization is $\Delta t = 0.001$ for the numerical integration. To evaluate the generalization ability, we inject step inputs

$$u(t) = \bar{u}$$

for some $\bar{u} \ge 0$.

Fig. 7 shows the time response of the reconstructed trajectory $\hat{x}(t)$ for $u(t) = 0$, which is close to the original one in Fig. 4. In particular, Figs. 8 and 9 magnify the responses for $0 \le t \le 0.1$. As expected from the observation in the previous paragraph, only the trajectory of the **CoDA** approximates the original one well for this fast time scale. The errors corresponding to the several inputs

$$E := \int_0^\tau \|x(t) - \hat{x}(t)\|^2 dt \tag{22}$$

are given in Table III. The **CoDA** yields smaller errors when the input energy is small.

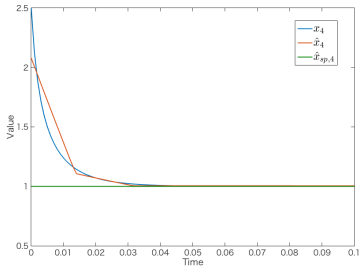

Fig. 8. Transient behavior of x_4 for $0 \leq t \leq 0.1$

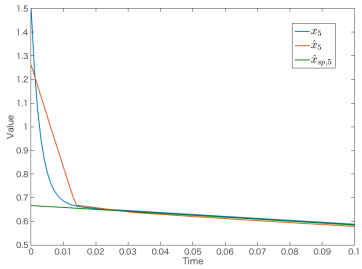

Fig. 9. Transient behavior of x_5 for $0 \leq t \leq 0.1$

VI. Conclusion

In this paper, we proposed a practical nonlinear model reduction method. The proposed **CoDA** is implemented as design of a deep autoencoder of the noise response data. The proposed method has the following important features:

1) It is applicable to nonlinear systems as far as their noise response data are available,
2) the control theoretic optimization criterion is considered,
3) it requires only the use of algorithms that are computationally efficient,
4) and it can achieve a performance beyond the limit of the reduction by linear projections.

The numerical example in Section V suggests that this direction is promising. Since deep learning is one of the most actively investigated research topics in various fields, there are numerous powerful techniques to enhance the learning efficiency [11], [12]. These techniques will surely enhance the practical usefulness of the proposed method. Furthermore, the **CoDA** inherently possesses some properties desirable for recent model reduction applications [2], [19]:

- the layered neural network automatically gives a *hierarchical model reduction*, and
- we can use regularization to impose a connection topology to perform *structure preserving model reduction*.

In deep learning, stacked autoencoders are used as the pretraining for regression, classification. Therefore, deep learning based on the **CoDA** will be useful to reinforced learning, fault detection, system identification, and so on, taking the controllability into account. Currently, these directions are under investigation by applying the proposed method to large-scale nonlinear network systems.

TABLE III

Comparison of error norm in Eq. (22) ($\times 10^{-3}$)

$\bar{u}$	0	0.1	0.5	1.0
CoDA	1.4242	1.6428	7.5407	29.1860
Singular perturbation	7.6114	7.6117	7.6130	7.6145

Acknowledgement

The author thanks Mr. Yuji Nagasawa of Kyoto University for his implementation of the deep learning algorithm. This work was in part supported by JSPS KAKENHI Grant Number 26289130.

References

[1] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, may 2011.
[2] T. Ishizaki, K. Kashima, J.-i. Imura, and K. Aihara, "Model reduction and clusterization of large-scale bidirectional networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 48–63, 2013.
[3] B. Besselink, N. van de Wouw, J. M. A. Scherpen, and H. Nijmeijer, "Model reduction for nonlinear systems by incremental balanced truncation," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2739–2753, oct 2014.
[4] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.
[5] A. Astolfi, "Model reduction by moment matching for linear and nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2321–2336, oct 2010.
[6] J. Scherpen, "Balancing for nonlinear systems," *Systems & Control Letters*, vol. 21, no. 2, pp. 143–153, aug 1993.
[7] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*, 2nd ed., ser. Cambridge monographs on mechanics. Cambridge University Press, 2012.
[8] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM Journal on Numerical Analysis*, vol. 40, no. 2, pp. 492–515, 2002.
[9] K. Kashima, "Noise response data reveal novel controllability Gramian for nonlinear network dynamics," *Scientific Reports*, vol. 6, p. 27300, jun 2016.
[10] ——, "Optimality of simulation-based nonlinear model reduction: Stochastic controllability perspective," in *2016 American Control Conference (ACC)*. IEEE, jul 2016, pp. 7243–7248.
[11] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.
[12] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
[13] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, 2nd ed., ser. Graduate texts in mathematics. Springer, 1998, no. 113.
[14] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.
[15] S. Lall, J. E. Marsden, and S. Glavaški, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *International Journal of Robust and Nonlinear Control*, vol. 12, no. September 2000, pp. 519–535, 2002.
[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief bets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, jul 2006.
[17] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, may 2013, pp. 8595–8598.
[18] P. V. Kokotović, H. K. Khalil, and J. J. O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*, ser. Classics in applied mathematics. Society for Industrial and Applied Mathematics, 1999, no. 25.
[19] T. Ishizaki, K. Kashima, A. Girard, J. I. Imura, L. Chen, and K. Aihara, "Clustered model reduction of positive directed networks," *Automatica*, vol. 59, pp. 238–247, 2015.