Simultaneous Identification of Nonlinear Dynamics and State Distribution using Jensen-Shannon Divergence

Kenji Kashima, *Senior Member, IEEE,* Moe Watanabe, Ichiro Maruta, *Member, IEEE*

***Abstract*— In this paper, we newly formulate and solve simultaneous identification problem of nonlinear dynamics and state distribution. This problem is practically useful in many realistic situations, but it has not attracted much attention from system identification community. From a mathematical point of view, estimation of state distribution is represented as a regularization in terms of the Jensen-Shannon divergence. An important feature of this formulation is its equivalence to construction of *generative models*, whose recent progress is one of the most important achievements in machine learning community. In view of this, we propose an adversarial learning approach, standard technique for generative model construction, to the aforementioned identification problem, and verify its effectiveness through numerical simulation.**

I. Introduction

System identification is a field that has been actively researched for a long time [1], [2], [3]. In this field, the traditional focus has been on the use of classical statistical methods like maximum likelihood estimation, and the difficulty of resulting optimization problems has been a significant issue in recent years [4]. Among these recent efforts, researches on methods introduced from machine learning community have been actively conducted, such as estimation of impulse response using kernel methods [5], [6], identification of nonlinear system using Gaussian process state space models [7], [8], and refocusing to neural network based method [9], [10]. Furthermore, system identification still has considerable room to introduce ideas.

Roughly speaking, data interpolation implemented by sophisticated regularization techniques is the central idea of machine learning. Therefore, it is not reasonable to utilize the obtained model outside the interpolated region. Possible remedies for small-data cases are to

(A1) extrapolate the data by means of prior knowledge of the system to be modeled, or

(A2) utilize the model only in the guaranteed domain.

A typical example of (A1) is the linear system identification. In this case, the model obtained by the local data can be effective all over the state space. It may also be useful to employ specific domain knowledge case-by-case to identify nonlinear system, e.g., gray-box modeling. The focus of this paper is put on (A2); see also Section II-A. Unfortunately, the characterization of the interpolated region is not necessarily trivial. This is mainly because

(B1) the domain does not necessarily have mathematically simple representation,

(B2) for system identification in the state-space representation, there is a degree of freedom of coordinate transform of state variables.

In this paper, in order to investigate such a situation in a mathematically rigorous fashion, we newly formulate *simultaneous identification problem of nonlinear dynamics and state distribution*, as detailed in Section II. This formulation using the Jensen-Shannon divergence has the same structure as construction of specific probabilistic models that are referred to as generative models in machine learning community [11]. It can be noted that adversarial learning approach to construct generative models brought about lots of efficient algorithms and wide variety of applications [12], [13], [14], [15]. This rich framework, however, has not been fully explored in controls community, to the best of the authors' knowledge [16]. In view of this, in Section III, the aforementioned system identification is performed by the adversarial learning approach. Numerical examples are illustrated in Section IV to verify the usefulness of the proposed method, and we offer concluding remarks in Section V.

Notation: For a vector $y \in \mathbb{R}^p$, $\|y\| := (y^\top y)^{1/2}$ denotes the Euclidean norm. The identity map is denoted by $\mathrm{Id}(x) := x$. We say $x \sim \mathbb{P}$ when x is a random variable obeying the probability measure $\mathbb{P}$. The normal distribution with mean vector μ and covariance matrix $\Sigma \succeq 0$ is denoted by $\mathcal{N}(\mu, \Sigma)$. For two probability density functions p and q, we introduce the Kullback-Leibler divergence

$$D_{\mathrm{KL}}(p\|q) := \int p(x) \log \frac{p(x)}{q(x)} dx \tag{1}$$

and the Jensen-Shannon divergence

$$D_{\mathrm{JS}}(p\|q) := \frac{1}{2}\left\{ D_{\mathrm{KL}}\left(p \middle\| \frac{p+q}{2}\right) + D_{\mathrm{KL}}\left(q \middle\| \frac{p+q}{2}\right)\right\}. \tag{2}$$

We abuse the notation $D_{\mathrm{JS}}(\mathbb{P}\|\mathbb{Q})$ for two probability measures $\mathbb{P}$, $\mathbb{Q}$ assuming the existence of the probability density functions. The expectation is represented by $\mathbb{E}$, whose probability measure is omitted when it is clear from the context.

II. Problem Formulation

A. Motivation

Suppose that the system to be modeled has the following discrete-time state-space realization

$$\hat{x}(k+1) = \hat{f}(\hat{x}(k)),\ \hat{x}(0) = \hat{x}_{\mathrm{i}},\ y(k) = \hat{g}(\hat{x}(k)) \tag{3}$$

K. Kashima and M. Watanabe are with Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. I. Maruta is with Graduate School of Engineering, Kyoto University, Kyoto, 606-8501, Japan. (e-mail: kk@i.kyoto-u.ac.jp, watanabe.moe@bode.amp.i.kyoto-u.ac.jp, maruta@kuaero.kyoto-u.ac.jp)

where the dimension of variables are $\hat{x} \in \mathbb{R}^{\hat{n}}$, $y \in \mathbb{R}^{n_y}$. For simplicity of the presentation, we fix the time length of data as an positive integer ℓ. Then, we refer to the solution sequence to this difference equation as

$$\phi(\hat{f}, \hat{g}, \hat{x}_{\rm i}) := \begin{bmatrix} \hat{g}(\hat{x}_{\rm i}) \\ \hat{g}(\hat{f}(\hat{x}_{\rm i})) \\ \vdots \\ \hat{g}(\underbrace{\hat{f} \dots \hat{f}}_{\ell-1}(\hat{x}_{\rm i})) \end{bmatrix} \in \mathbb{R}^{\ell n_y}. \tag{4}$$

The goal of standard system identification problem is to construct

$$x(k+1) = f(x(k)),\ x(0) = x_{\rm i},\ y(k) = g(x(k)), \tag{5}$$

and $\pi : \mathbb{R}^{\hat{n}} \to \mathbb{R}^n$, where the dimension of variables are $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n_y}$, such that

$$\phi(\hat{f}, \hat{g}, \hat{x}_{\rm i}) \approx \phi(f, g, \pi(\hat{x}_{\rm i})) \tag{6}$$

in a suitable sense. For this purpose, we assume the availability of samples

$$\hat{y}_{\rm o}^{(s)} := \phi(\hat{f}, \hat{g}, \hat{x}_{\rm i}^{(s)}) + \epsilon^{(s)} \tag{7}$$

for some $(\hat{x}_{\rm i}^{(s)})_{s=1}^N$ with the independent Gaussian observation noise $\epsilon^{(s)} \sim \mathcal{N}(0, \sigma^2 I)$. The main idea of the prediction error method is to minimize

$$J_{\rm pem}(f, g, (x_{\rm i}^{(s)})_{s=1}^N) := \sum_s \|\hat{y}_{\rm o}^{(s)} - \phi(f, g, x_{\rm i}^{(s)})\|^2 \tag{8}$$

over given function classes.

Let us consider a realistic situation where $\hat{x}_{\rm i}^{(s)}$ *distributes only on a small domain* $\hat{\Omega} \subset \mathbb{R}^{\hat{n}}$, possibly in a non-uniform manner. In such a case, the available data does not contain information about the trajectories starting from outside of $\hat{\Omega}$. Suppose that, based on the minimization of (8), we obtain some $f,\ g$ and π. Then, although (6) may hold for $\hat{x}_{\rm i} \in \hat{\Omega}$, we cannot expect *extrapolation*. That is, $\phi(f, g, \pi(\hat{x}_{\rm i}))$ with $\hat{x}_{\rm i} \notin \hat{\Omega}$ should be a meaningless sequence that has nothing to do with the system to be modeled. See also Section IV-B for details. This observation motivates the present work. It should be emphasized that the goal of this paper is not the extrapolation, but to *construct model (5) as well as a simple characterization of initial states* $x_{\rm i}$ *such that* $\phi(f, g, x_{\rm i})$ *captures suitable behavior of (3).*

B. Proposed framework

Toward this goal, suppose that the system to be modeled has the state space realization (3) with probability distribution

$$\hat{x}_{\rm i} \sim \hat{\mathbb{P}}_{\rm i} \tag{9}$$

where $\hat{\mathbb{P}}_{\rm i}$ is a probability distribution over $\mathbb{R}^{\hat{n}}$. We assume that we can utilize the same noisy data as (7). The only difference is that $\hat{x}_{\rm i}^{(s)}$ is an i.i.d. sample according to $\hat{\mathbb{P}}_{\rm i}$.

Although the information of $\hat{\mathbb{P}}_{\rm i}$ has been rarely considered in system identification literature, it is practically useful in many realistic situations. For example, the dynamics of objects captured by camera or radar is required for autonomous driving [17]. In this type of application, only a large amount of short sequences can be used as training data, and the initial state distribution, which corresponds to the information of the objects entering into the field, is highly useful.

Remark 1. *It is more common in system identification literature to assume that a single long sequential data is available. The proposed framework can cover such a situation by dividing the sequence into appropriate lengths. In this case, if the system is ergodic,* $\hat{\mathbb{P}}_{\rm i}$ *will correspond to the distribution of the state in the stationary normal operation, and suggest a region where the model can be trusted.*

A naive approach to extract the information of $\hat{\mathbb{P}}_{\rm i}$ from the observation sequences $\left(\hat{y}_{\rm o}^{(s)}\right)_{s=1}^N$ may be to minimize (8) first, and then to characterize the probability distribution of $\{x_{\rm i}^{(s)}\}$. However, this is not an easy task at all due to the reason (B1), (B2) stated in the Introduction. In particular, because the use of a highly expressive and redundant model like a neural network is assumed here, the coordinate system used in state space can easily become overly complicated. For such a situation, we need complicated models to represent the distribution of $\left(\hat{y}_{\rm o}^{(s)}\right)_{s=1}^N$, for which explicit characterization of the likelihood function is not available, and the classical maximum likelihood estimation is not applicable [18], [19], [20]. In addition, the ease of sampling from the initial state distribution is vital in the use of the model for particle filters, which are extensively used in autonomous driving, and for verification in simulations.

C. Generative model formulation

In what follows, we take a converse approach: First, fix a desirable (e.g., sample efficient) probability measure $\mathbb{P}_{\rm i}$ on $\mathbb{R}^n$, then construct a state space model whose initial state distribution conforms to it. To put it precisely, we utilize the degree of freedom of state variables in (5) to make $J_{\rm pem}$ in (8) small with $\{x_{\rm i}^{(s)}\} \sim \mathbb{P}_{\rm i}$. Finally, the obtained model is given by (5) combined with

$$x_{\rm i} \sim \mathbb{P}_{\rm i}. \tag{10}$$

For notational simplicity, we hereafter denote

$$\hat{y}_{\rm o}(\hat{x}_{\rm i}) := \phi(\hat{f}, \hat{g}, \hat{x}_{\rm i}). \tag{11}$$

Now, we are in the position to formulate several types of problems as unsupervised learning where (6) is interpreted differently.

Problem 1 (Generative model for output sequences)**.** *Find a map* $G : \mathbb{R}^n \to \mathbb{R}^{\ell n_y}$ *that minimizes*

$$J_1(G) := D_{\rm JS}\left(\hat{y}_{\rm o}(\hat{x}_{\rm i})|_{\hat{x}_{\rm i} \sim \hat{\mathbb{P}}_{\rm i}} \middle\| G(x_{\rm i})|_{x_{\rm i} \sim \mathbb{P}_{\rm i}}\right). \tag{12}$$

□

In this formulation, $G(x_{\rm i})$ induces a probability distribution over the space of output sequences $\mathbb{R}^{\ell n_y}$. Then, J_1 evaluates the gap of distributions over sequences. Once we find G that makes $J_1(G)$ small, we can mimic $\phi(\hat{f}, \hat{g}, \hat{x}_{\rm i})|_{\hat{x}_{\rm i} \sim \hat{\mathbb{P}}_{\rm i}}$

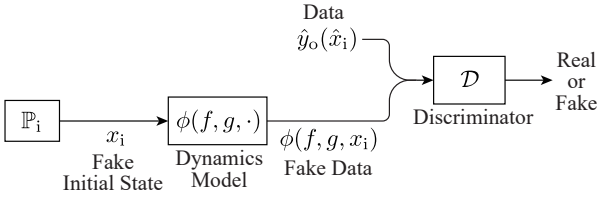

Fig. 1. Illustration of Problem 2, where f and g are tuned to minimize the discrimination rate, while $\mathcal{D}$ is devised to maximize the discrimination rate.

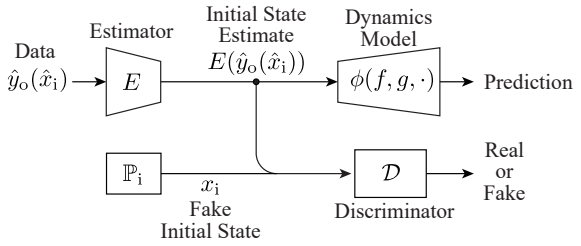

Fig. 2. Illustration of Problem 3, where E, f, and g are tuned to minimize the prediction error and discrimination rate, while $\mathcal{D}$ is devised to maximize the discrimination rate.

by simply sampling $x_{\rm i} \sim \mathbb{P}_{\rm i}$ and substituting it to the signal generator G. This is why such a model is called a generative model.

Next, imposing specific causality structure for G according to (3), we obtain the following:

Problem 2 (System Identification with initial probability density estimation)**.** *Find f, g that minimize*

$$J_2(f,g) := D_{\rm JS}\left(\hat{y}_{\rm o}(\hat{x}_{\rm i})|_{\hat{x}_{\rm i}\sim\hat{\mathbb{P}}_{\rm i}} \middle\| \phi(f,g,x_{\rm i})|_{x_{\rm i}\sim\mathbb{P}_{\rm i}}\right). \quad (13)$$

□

In comparison to Problem 1, $x_{\rm i}$ corresponds to the initial states for the constructed model; see Fig. 1

It should be noted that (12) and (13) evaluates the gap of probability distributions only. This means that the results do not provide relationship between individual initial states, e.g., π in (6). It is practically useful to compute the initial state from an output when the obtained model is used for state filtering or a soft sensor. We incorporate such an idea as shown in Fig 1.

Problem 3 (Unified design of model, density, and state-estimator)**.** *Given $\gamma > 0$, find f, g, and $E : \mathbb{R}^{\ell n_y} \to \mathbb{R}^n$ that minimizes*

$$J_3(E,f,g) := J_3^{\rm rec}(E,f,g) + \gamma J_3^{\rm gm}(E) \quad (14)$$

where

$$J_3^{\rm rec}(E,f,g) := \mathbb{E}_{\hat{x}_{\rm i}\sim\hat{\mathbb{P}}_{\rm i}}\|\hat{y}_{\rm o}(\hat{x}_{\rm i}) - \phi(f,g,E(\hat{y}_{\rm o}(\hat{x}_{\rm i})))\|^2 \quad (15)$$

and

$$J_3^{\rm gm}(E) := D_{\rm JS}\left(E(\hat{y}_{\rm o}(\hat{x}_{\rm i}))|_{\hat{x}_{\rm i}\sim\hat{\mathbb{P}}_{\rm i}} \middle\| \mathbb{P}_{\rm i}\right). \quad (16)$$

The first term evaluates the *reconstruction error*. Artificial neural networks having such a structure are called *autoencoders* [11], [21]. Hence, the estimator E becomes a map from a sequence to an initial state that reconstructs the observed sequence through the obtained model. In other words, the composite map $E(\hat{y}_{\rm o}(\cdot))$ plays the role of π in (6). The second term is to obtain a generative model for initial states of the model.

III. Adversarial learning approach

Unlike the classical formulation as maximum likelihood estimation based on the Kullback-Leibler divergence, Problems 1–3, which are based on the Jensen-Shannon divergence, are tractable even for the model with latent variables. For example, Problem 3 has the same structure as *adversarial autoencoder* design [14], where the estimator, dynamical model and initial state play a role of an encoder, decoder, and feature vector, respectively. Therefore, standard adversarial learning techniques can be employed:

Theorem 1. *For $\mathcal{D} : \mathbb{R}^n \to [0,1]$, define the cross entropy*

$$J^{\rm ce}(E,\mathcal{D}) := \mathbb{E}_{\hat{x}_{\rm i}\sim\hat{\mathbb{P}}_{\rm i}}[\log \mathcal{D}(E(\hat{y}_{\rm o}(\hat{x}_{\rm i})))] + \mathbb{E}_{x_{\rm i}\sim\mathbb{P}_{\rm i}}[\log(1-\mathcal{D}(x_{\rm i}))] \quad (17)$$

and

$$\mathcal{L}(E,f,g,\mathcal{D}) := J_3^{\rm rec}(E,f,g) + \frac{\gamma}{2} J^{\rm ce}(E,\mathcal{D}). \quad (18)$$

Let

$$\mathcal{D}^\star(E) := \arg\max_{\mathcal{D}} \; J^{\rm ce}(E,\mathcal{D}). \quad (19)$$

Then,

$$J_3(E,f,g) = \mathcal{L}(E,f,g,\mathcal{D}^\star(E)). \quad (20)$$

Proof: We can verify that the maximizer $\mathcal{D}^\star(E)$ is given by the density ratio

$$\mathcal{D}^\star(E) := \frac{p(x)}{p(x)+q(x)} \quad (21)$$

where p and q are probability density functions[1] for $\mathbb{P}_{\rm i}$ and $E(\hat{y}_{\rm o}(\hat{x}_{\rm i}))|_{\hat{x}_{\rm i}\sim\hat{\mathbb{P}}_{\rm i}}$. By substituting this into $J^{\rm ce}(E,\mathcal{D})$, we have

$$J^{\rm ce}(E,\mathcal{D}^\star(E)) = 2J_3^{\rm gm}(E) - 2\log 2. \quad (22)$$

This completes the proof. □

The newly introduced $\mathcal{D}$ can be interpreted as a discriminator that attempts to output the degree of belief whether the input data is real ($E(\hat{y}_{\rm o}(\hat{x}_{\rm i}))$) or fake ($x_{\rm i} \sim \mathbb{P}_{\rm i}$). Note that $\mathcal{D}^\star(E) = 0.5$ in (21) and $J^{\rm ce}(E,\mathcal{D}^\star(E)) = -2\log 2$ in (22) if the two probability measures are identical. This result motivates us to solve Problem 3 via the min-max problem

$$\min_{E,f,g}\max_{\mathcal{D}} \mathcal{L}(E,f,g,\mathcal{D}). \quad (23)$$

In summary we iterate

1) For fixed E, we update $\mathcal{D}$ so that $J^{\rm ce}(E,\mathcal{D})$ is maximized, and
2) For fixed $\mathcal{D}$, we update (E,f,g) so that $\mathcal{L}(E,f,g,\mathcal{D})$ decreases.

[1]We implicitly assumed these two measures are absolutely continuous.

For the implementation, a stochastic gradient method is adopted, where the expectation with respect to $\hat{x}_\mathrm{i} \sim \hat{\mathbb{P}}_\mathrm{i}$ and $x_\mathrm{i} \sim \mathbb{P}_\mathrm{i}$ in J_3^{rec} and J^{ce} are approximated by random sampling from noisy observation $\left(\hat{y}_\mathrm{o}^{(s)}\right)_{s=1}^{N}$ in (7) and random realization from $\mathbb{P}_\mathrm{i}$, respectively. Suppose E, f, g and $\mathcal{D}$ are parametrized as E_θ, f_θ, g_θ, and $\mathcal{D}_{\theta_d}$ with parameter vectors $\theta \in \Theta$ and $\theta_d \in \Theta_d$. Actual algorithm is given in Algorithm 1.

Algorithm 1 Unified design of model, density, and state-estimator (Problem 3) by adversarial learning

Require: data $\left(\hat{y}_\mathrm{o}^{(s)}\right)_{s=1}^{N}$, fake initial state distribution $\mathbb{P}_\mathrm{i}$, initial value for parameter θ and θ_d

Ensure: designed encoder E_θ, dynamics model f_θ and g_θ, discriminator $\mathcal{D}_{\theta_\mathrm{d}}$

1: **repeat**

2: **for** $i = 1, \ldots, m$ **do**

3: Sample minibatch of N_B data $\left(\hat{y}_\mathrm{o}^{(s_k)}\right)_{k=1}^{N_\mathrm{B}}$ from $\left(\hat{y}_\mathrm{o}^{(s)}\right)_{s=1}^{N}$.

4: Sample minibatch of N_B fake initial states $\left(x_\mathrm{i}^{(k)}\right)_{k=1}^{N_\mathrm{B}}$ from $\mathbb{P}_\mathrm{i}$.

5: Update θ_d by ascending stochastic gradient

$$\nabla_{\theta_d}\left[\frac{1}{N_\mathrm{B}}\sum_{k=1}^{N_\mathrm{B}}\left\{\log \mathcal{D}_{\theta_d}\left(E_\theta\left(\hat{y}_\mathrm{o}^{(s_k)}\right)\right) + \log\left(1-\mathcal{D}_{\theta_d}\left(x_\mathrm{i}^{(k)}\right)\right)\right\}\right].$$

6: **end for**

7: Sample minibatch of N data $\left(\hat{y}_\mathrm{o}^{(s_k)}\right)_{k=1}^{N_\mathrm{B}}$.

8: Update θ by descending stochastic gradient

$$\nabla_{\theta}\left[\frac{1}{N_\mathrm{B}}\sum_{k=1}^{N_\mathrm{B}}\left\{\left\|\hat{y}_\mathrm{o}^{(s_k)} - \phi\left(f_\theta, g_\theta, E_\theta\left(\hat{y}_\mathrm{o}^{(s_k)}\right)\right)\right\|^2 + \frac{\gamma}{2}\log \mathcal{D}_{\theta_d}\left(E_\theta\left(\hat{y}_\mathrm{o}^{(s_k)}\right)\right)\right\}\right].$$

9: **until** θ converges

Although this iteration does not necessarily converge to the saddle point of the min-max problem (23), various heuristics for this purpose have been proposed in the machine learning literature.

IV. Numerical simulation

A. Simulation setting

In order to focus on the main topic of the paper, which is the density estimation, we take $\hat{g} := \mathrm{Id}$, i.e., the state is directly observable. However, this information $\hat{g} = \mathrm{Id}$ is not utilized explicitly in the following modeling procedure. Let us consider the following continuous-time system on $\mathbb{R}^2$:

$$\frac{d}{dt}\|x(t)\| = \begin{cases} 1-\|x(t)\|, & \text{if } \|x(t)\| \geq 0.5, \\ -\|x(t)\|, & \text{otherwise,} \end{cases} \quad (24)$$

$$\frac{d}{dt}\arg(x(t)) = 1. \quad (25)$$

TABLE I

Model structure in numerical example

Model	Structure	# parameters
E_θ	Input (400) → Identity (2) → ReLU (5) → ReLU (5) → ReLU (2) → Output (2)	865
f_θ	Input (2) → ReLU (8) → ReLU (8) → ReLU (2) → Output (2)	120
g_θ	Input (2) → ReLU (4) → ReLU (5) → Output (2)	55
$\mathcal{D}_{\theta_d}$	Input (2) → ReLU (8) → ReLU (64) → ReLU (8) → ReLU (2) → Sigmoid (1) → Output (1)	1141

It can be easily seen that all the trajectories are attracted to the origin (fixed point) or the unit circle (limit cycle). The vector field $\hat{f}$ in (3) is given by the time-discretization of this continuous-time one with time step $\Delta t = 0.1$; see Fig. 3. The probability distribution $\hat{\mathbb{P}}_\mathrm{i}$ of the initial state is the uniform distribution over

$$\hat{\Omega} := \left\{\hat{x}_\mathrm{i} \in \mathbb{R}^2 : 0.5 \leq \|\hat{x}_\mathrm{i}\| \leq 1.5\right\}. \quad (26)$$

The observation noise is taken to be $\epsilon \sim \mathcal{N}(0, 0.035^2 \cdot I)$; see Fig. 4 for sample observation (7). Note that all the trajectories start from inside $\hat{\Omega}$.

For implementing the lines 5 and 8 in Algorithm 1, Adam optimizer in PyTorch package is used [22]. The models E_θ, f_θ, g_θ, and $\mathcal{D}_{\theta_d}$ are implemented by neural networks which have 865, 120, 55, and 1141 parameters, respectively. The structures of the models are summarized in Table I. The numbers show the number of units in the layers.

B. Prediction error method

First, we execute the standard prediction error method to recall our motivation of this paper. By minimizing $J_3^{\mathrm{rec}}(E, f, g)$ we obtained f_{pem}, g_{pem} and E_{pem} in order to achieve (6). Fig. 5 (blue) shows $E_{\mathrm{pem}}(\hat{y}_\mathrm{o}(\hat{x}_\mathrm{i}))$ for $\hat{x}_\mathrm{i} \sim \hat{\mathbb{P}}_\mathrm{i}$. Note that this distribution characterizes the set of suitable initial states for the obtained model. To see this, $\phi(f_{\mathrm{pem}}, g_{\mathrm{pem}}, x_\mathrm{i})$ are shown in Fig. 6. The blue line, which is for x_i in the blue region in Fig. 5, starts[2] inside $\hat{\Omega}$ and captures the convergence to the stable limit cycle appropriately. On the other hand, the red line, for x_i out of the blue region, does not start from $\hat{\Omega}$. In particular, the inner trajectory converges to the limit cycle due to the absence of the training data, which contradicts to the attraction to the origin in Fig. 3. The latter case indicates the failure of extrapolation. It should be emphasized that the probability distribution over the blue region in Fig. 5 is informative, but not trivial to characterize accurately and varies depending on initialization and sotchasticity involved in the algorithm.

C. Proposed method

Next, we attempted to solve Problem 2 via adversarial learning similar to Algorithm 1. However, in almost all the

[2]Since $g_{\mathrm{pem}}(x) \neq x$, $x_\mathrm{i} \neq y(0)$ in each trajectory.

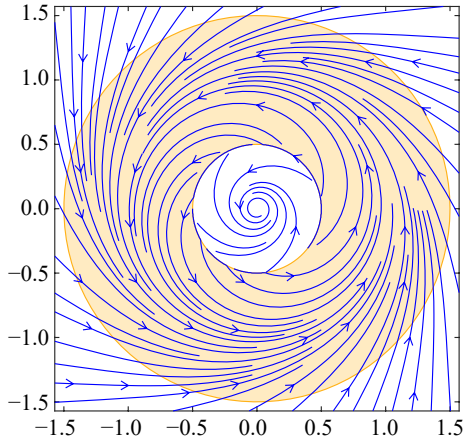

Fig. 3. Vector field of the dynamics $\hat{f}$ (stream lines) and the area where the initial state exists $\hat{\Omega}$ (colored area) to be modeled.

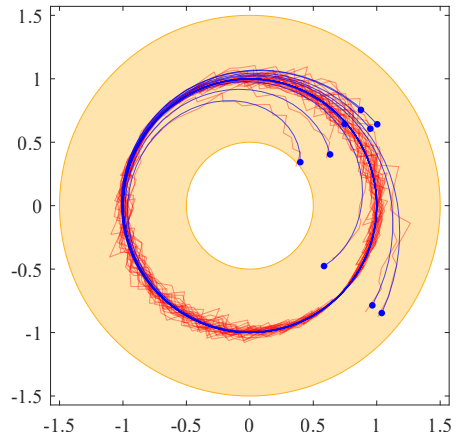

Fig. 4. Sample trajectories: original $\hat{y}_{\mathrm{o}}(\hat{x}_{\mathrm{i}}) = \phi(\hat{f}, \hat{g}, \hat{x}_{\mathrm{i}})$ with $\hat{x}_{\mathrm{i}} \in \hat{\Omega}$ (blue); noisy observation $\hat{y}_{\mathrm{o}}(\hat{x}_{\mathrm{i}}) + \epsilon$ (red).

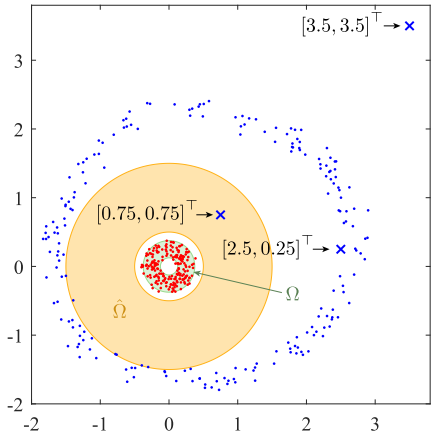

Fig. 5. Scattering plot of encoded initial states: PEM $E_{\mathrm{pem}}(\hat{y}_{\mathrm{o}}(\hat{x}_{\mathrm{i}}))$ for $\hat{x}_{\mathrm{i}} \sim \hat{\mathbb{P}}_{\mathrm{i}}$ (blue), and proposed method $E^{\star}(\hat{y}_{\mathrm{o}}(\hat{x}_{\mathrm{i}}))$ for $\hat{x}_{\mathrm{i}} \sim \hat{\mathbb{P}}_{\mathrm{i}}$ (red).

cases we have tried, we failed to train the discriminator $\mathcal{D}$ over high dimensional space $\mathbb{R}^{n_y \ell}$.

Next, we move on to Problem 3 with $\gamma = 1$. For our proposed method, $\mathbb{P}_{\mathrm{i}}$ is set to be the uniform distribution over

$$\Omega := \frac{1}{4}\hat{\Omega} := \left\{ x_{\mathrm{i}} \in \mathbb{R}^2 : 0.125 \leq \|x_{\mathrm{i}}\| \leq 0.375 \right\} \tag{27}$$

for intuitive understanding of the results. In this case,

$$f(x) := \frac{1}{4}\hat{f}(4x), \;\; g(x) := 4x, \tag{28}$$

and E such that

$$E(\hat{y}_{\mathrm{o}}(\hat{x})) = \frac{1}{4}\hat{x} \tag{29}$$

is an optimal model for Problem 3. Algorithm 1 is executed with the vector field f and g initialized by f_{pem} and g_{pem} obtained above. The estimator E is initialized as the moving average, that is, $E(\hat{y}_{\mathrm{o}})$ is the arithmetic mean vector of the first 10 vectors of each observation sequence $\hat{y}_{\mathrm{o}}$.

Concerning the discriminator learning, $\mathcal{D}^{\star}(x)$ is close to 0.5 uniformly on Ω, and $J^{\mathrm{ce}}(E^{\star}, \mathcal{D}^{\star}) \approx -1.34$ is close to $-2\log 2$, as expected. Actually, the scatter plot of random samples according to $E^{\star}(\hat{y}_{\mathrm{o}}(\hat{x}_{\mathrm{i}}))|_{\hat{x}_{\mathrm{i}} \sim \hat{\mathbb{P}}_{\mathrm{i}}}$ depicted in Fig. 5 (red) looks a good approximation of $\mathbb{P}_{\mathrm{i}}$, as desired.

Fig. 7 shows $\phi(f^{\star}, g^{\star}, x_{\mathrm{i}})$ with x_{i} sampled according to $\mathbb{P}_{\mathrm{i}}$. The similarity between Figs. 4 and 7 suggests that $f^{\star}$, $g^{\star}$ with $\mathbb{P}_{\mathrm{i}}$ is a good generative model of output sequences, which implies $J_{\mathrm{JS}}(f^{\star}, g^{\star})$ in (13) is small. Finally, Fig. 8 shows $y_{\mathrm{o}} = \phi(\hat{f}, \hat{g}, x_{\mathrm{i}}) + \epsilon$ and $\phi(f^{\star}, g^{\star}, E^{\star}(\hat{y}_{\mathrm{o}}))$, where $x_{\mathrm{i}} \in \hat{\Omega}$. This result shows $E^{\star}$ successfully estimates the corresponding initial state.

It should be noted that similar results were obtained for some other low-dimensional nonlinear systems with their simple initial state distributions as well.

V. Conclusions

In this paper, we have proposed a novel framework of system identification based on the idea of generative models and adversarial learning. The proposed method enables us to identify not only the vector field, but also the probability density for the initial states, which is useful to estimate a state region on which the obtained model is reliable. In addition to comprehensive numerical study using complicated and realistic systems, extension to

- systems with external inputs,
- continuous-time system identification using highly expressive models from a viewpoint of [23]
- other metric such as Wasserstein distance [24]

and relation to

- optimal transport [25], [26] and
- covariance assignment [27]

are currently under investigation.

Acknowledgment

This work was supported in part by JSPS KAKENHI under Grant Number JP18H01461.

References

[1] K. Åström and P. Eykhoff, "System identification — a survey," *Automatica*, vol. 7, no. 2, pp. 123 – 162, 1971.

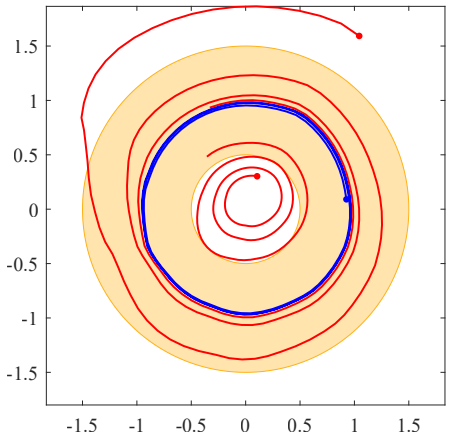

Fig. 6. Trajectories generated by PEM model $\phi(f_{\text{pem}}, g_{\text{pem}}, x_{\text{i}})$: for $x_{\text{i}} = [2.5, 0.25]^{\mathsf{T}}$ (blue), and for $x_{\text{i}} = [3.5, 3.5]^{\mathsf{T}}, [0.75, 0.75]^{\mathsf{T}}$ (red).

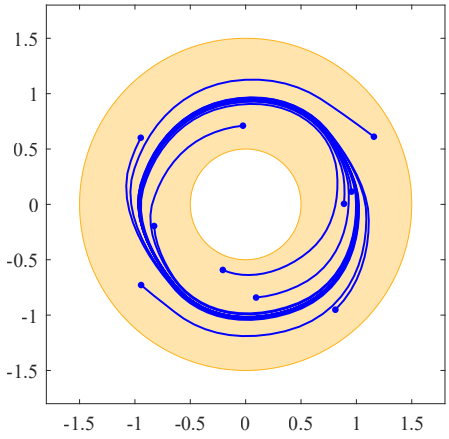

Fig. 7. Fake trajectory generation: $\phi(f^\star, g^\star, x_{\text{i}})$ for fake initial state $x_{\text{i}} \sim \mathbb{P}_{\text{i}}$.

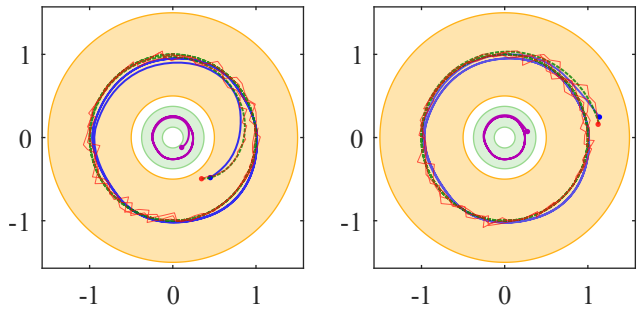

Fig. 8. Trajectory reconstruction: original $\hat{y}_{\text{o}}(\hat{x}_{\text{i}}) = \phi(\hat{f}, \hat{g}, \hat{x}_{\text{i}})$ with $\hat{x}_{\text{i}} \in \hat{\Omega}$ (dashed green); noisy observation $y_n = \hat{y}_{\text{o}}(\hat{x}_{\text{i}}) + \epsilon$ (thin red); model state $x = \phi(f^\star, \text{Id}, E^\star(y_n))$ (purple); reconstructed $g^\star(x)$ (blue).

[2] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice-Hall, 1999.

[3] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*, 2nd ed. New York: IEEE Press, 2012.

[4] L. Ljung, "On Convexification of System Identification Criteria," *Automation and Remote Control*, vol. 80, no. 9, pp. 1591–1606, 2019.

[5] L. Ljung, T. Chen, and B. Mu, "A shift in paradigm for system identification," *International Journal of Control*, vol. 93, no. 2, pp. 173–180, 2020.

[6] G. Pillonetto and G. D. Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81 – 93, 2010.

[7] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian Process State Space Models," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5309–5319.

[8] R. Frigola, Y. Chen, and C. E. Rasmussen, "Variational gaussian process state-space models," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 3680–3688.

[9] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[10] C. R. Andersson, A. H. Ribeiro, K. Tiels, N. Wahlström, and T. B. Schön, "Deep convolutional networks in system identification," *arXiv preprint arXiv:1909.01730v2*, 2019.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information science and statistics / series editors M. Jordan ... [et al.]. Springer, 2006.

[12] C. Li, K. Xu, J. Zhu, and B. Zhang, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 4089–4099, 2017.

[13] M. Mirza and S. Osindero, "Conditional generative adversarial nets," pp. 1–7, nov 2014.

[14] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," pp. 1–10, nov 2015.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, no. Ml, pp. 1–14, 2014.

[16] K. Shimizu, H. Nakada, and K. Kashima, "Feature extraction of internal dynamics of an engine air path system: Deep autoencoder approach," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 736–741, 2018.

[17] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, p. 1, 2014.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2672–2680.

[19] A. Grover, M. Dhar, and S. Ermon, "Flow-GAN: Combining maximum likelihood and adversarial learning in generative models," in *Thirty-Second AAAI Conference on Artificial Intelligence*, April 2018, pp. 3070–3076.

[20] K. Li and J. Malik, "Implicit maximum likelihood estimation," *arXiv preprint arXiv:1809.09087*, 2018.

[21] K. Kashima, "Nonlinear model reduction by deep autoencoder of noise response data," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, dec 2016, pp. 5750–5755.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[23] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 6571–6583.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 214–223.

[25] G. Peyré and M. Cuturi, "Computational optimal transport," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–206, 2019.

[26] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal steering of a linear stochastic system to a final probability distribution, part I," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1158–1169, may 2016.

[27] H. Fujioka and S. Hara, "State covariance assignment problem with measurement noise: a unified approach based on a symmetric matrix equation," *Linear Algebra and Its Applications*, vol. 203-204, no. C, pp. 579–605, 1994.