2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Neural Network-Based Nonlinear System Identification for Generating Stochastic Models with Distribution Estimation

Keito Yamada, Ichiro Maruta and Kenji Fujimoto

Abstract— **This paper proposes a nonlinear system identification method for constructing models that provide not only point estimates but also distribution. The method is based on a nonlinear system identification method using the concepts of bottleneck structured neural networks and subspace system identification, and further applies the concept of variational autoencoders. The validity of the proposed method is confirmed through numerical examples.**

I. Introduction

System identification is crucial in control system design when there is limited prior knowledge on the target system. While various identification methods exist for linear systems [1], nonlinear system identification remains an active area of research with techniques like Koopman Operator [2] and neural networks [3]. Among the linear system identification methods, the subspace identification method [4] is one of the most popular methods as it can directly obtain multi-input, multi-output state-space models. Building upon the subspace identification method and autoencoder concepts, the authors previously proposed an extended subspace identification method for nonlinear systems using neural networks [5].

The method performs well for nonlinear multi-input, multi-output systems of a practical scale and gives a model consisting of a state estimator and an output predictor that is particularly useful in model predictive control. Nonetheless, it is important to acknowledge that, akin to most nonlinear system identification techniques, this method does not yield explicit insights into the uncertainty of the predicted output and the state estimate caused by noise or disturbances. This is a notable consideration, especially given that there are control approaches utilizing such information, such as probabilistically constrained model predictive controller [6] and scenario-based approach [7].

On the other hand, probabilistic models are also frequently used in the field of machine learning. Variational Auto-Encoder (VAE) [8] is one of them, which estimates the distribution of latent variables based on variational inference using neural networks. So far, some research has been conducted on estimating dynamics using VAEs, and several methods have been proposed for estimating the state of time series data and the distribution of outputs by combining recurrent neural networks (RNN) [9], [10], [11], [12]. However, these methods require large amounts of sampling or recursive calculations to evaluate output uncertainties in multiple steps, which can be computationally time-consuming. This is particularly undesirable in applications such as model predictive control [13], where predictions must be made in a short period of time.

This work was supported by JSPS KAKENHI Grant Number JP20H02170.

K. Yamada, I. Maruta and K. Fujimoto is with Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University, Kyotodaigaku-katsura, Nishikyo Ward, Kyoto City, Kyoto, 615-8540, Japan yamada.keito.33w@st.kyoto-u.ac.jp, {maruta,fujimoto}@kuaero.kyoto-u.ac.jp

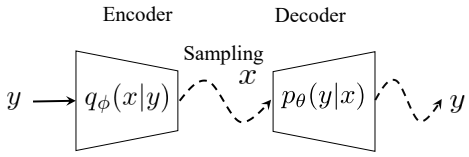


Fig. 1. Structure of VAE

In this paper, we propose a method for estimating the distribution of states and outputs by applying the VAE concept to the extended subspace identification method [5]. By using a probabilistic model, we can estimate the uncertainty of the output, and since recursion is not required, we can make predictions faster than in previous studies.

This paper is organized as follows. Section II provides an explanation of VAE. Section III briefly describes the nonlinear system identification proposed by the authors [5]. Section IV describes the nonlinear system identification with uncertainty, which is an application of the VAE concept, and Section V confirms the validness of the proposed method using numerical examples. Finally, Section VI presents the conclusions.

In this paper, the probability density at x of a random variable that follows the normal distribution with mean μ and covariance matrix Σ is denoted by $\mathcal{N}(x \mid \mu, \Sigma)$. The (i, j) element of matrix X is denoted by $(X)_{ij}$.

II. Variational Auto-Encoder (VAE) [8]

Let $y \in \mathbb{R}^{n_y}$ be an observed variable and $x \in \mathbb{R}^{n_x}$ be a latent variable. VAE models the distribution of x when y is observed via the encoder, and the distribution of y when x is given via the decoder. Fig. 1 illustrates the concept of VAE. In VAE, we can assume various types of the distribution of the decoder output, but in this paper we assume a normal distribution. In this case, the distribution of the decoder output can be expressed as

$$p_\theta(y \mid x) = \mathcal{N}(y \mid \mu_\theta(x), \Sigma_\theta(x)), \tag{1}$$

where μ_θ and Σ_θ are implemented by neural networks, and θ is a parameter of them. On the other hand, since the posterior distribution $p(x \mid y)$ is generally complex, variational inference is performed using the approximate posterior distribution $q_\phi(x \mid y)$. If the encoder outputs are normally distributed as

well as decoders, $q_\phi(x \mid y) = \mathcal{N}(x \mid \mu_\phi(y), \Sigma_\phi(y))$. ϕ denotes their parameters.

Given data $\{y_1, y_2, \dots, \}$ is independent and identically distributed (i.i.d.), it is desirable to estimate the parameters of VAE by maximizing the marginal log likelihood $\log p_\theta(y_1, y_2, \dots,) = \sum \log p_\theta(y_i)$. Unfortunately, $p_\theta(y)$ is often a complicated distribution that is difficult to compute, so we consider maximizing the evidence lower bound (ELBO) $\mathcal{L}(y; \theta, \phi)$ instead, which is defined as follows:

$$\begin{aligned}
\log p_\theta(y) &= \log \int p_\theta(y \mid x) p_\theta(x) dx \\
&= \log \int \frac{p_\theta(y \mid x) p_\theta(x)}{q_\phi(x \mid y)} q_\phi(x \mid y) dx \\
&= \log \mathbb{E}_{q_\phi(x|y)} \left[\frac{p_\theta(y \mid x) p_\theta(x)}{q_\phi(x \mid y)} \right] \\
&\geq \mathbb{E}_{q_\phi(x|y)} \left[\log \left(\frac{p_\theta(y \mid x) p_\theta(x)}{q_\phi(x \mid y)} \right) \right] \\
&= \mathbb{E}_{q_\phi(x|y)} \left[\log p_\theta(y \mid x) \right] \\
&\quad - \mathrm{D}_{\mathrm{KL}}[q_\phi(x \mid y) \parallel p_\theta(x)] \qquad (2) \\
&:= \mathcal{L}(y; \theta, \phi) \qquad (3)
\end{aligned}$$

The inequalities between the third and fourth lines are derived from Jensen's inequality. Also, $\mathrm{D}_{\mathrm{KL}}[\cdot \parallel \cdot]$ is the Kullback-Leibler (KL) divergence defined as

$$\mathrm{D}_{\mathrm{KL}}[p(x) \parallel q(x)] = \int p(x) \log \frac{p(x)}{q(x)} dx. \qquad (4)$$

ELBO can be decomposed into the first term related to the reconstruction error and the second regularization term as in (2). The first term is difficult to compute analytically, so it can be approximated by the Monte Carlo method as follows:

$$\begin{aligned}
\mathbb{E}_{q_\phi(x|y)} \left[\log p_\theta(y \mid x) \right] &= \frac{1}{K} \sum_{k=1}^{K} \left[\log p_\theta(y \mid x^{(k)}) \right], \\
x^{(k)} &\sim q_\phi(x \mid y) \qquad (5)
\end{aligned}$$

If $q_\phi(x \mid y)$ is normally distributed, then the sampling of $x^{(k)}$ can be easily performed as follows:

$$x^{(k)} = \mu_\phi(y) + \Sigma_\phi^{1/2}(y) \varepsilon^{(k)}, \qquad (6)$$

where the random variable $\varepsilon^{(k)} \sim \mathcal{N}(0, I)$ and $\Sigma^{1/2}$ represents the Cholesky decomposition of Σ. By replacing $x^{(k)}$ in (5) with the expression (6), it is possible to update the parameters of the neural network by back propagation. When the dataset is large, $K = 1$ is usually selected. On the other hand, the second term can be calculated analytically when $q_\phi(x \mid y)$, $p_\theta(x)$ are both normal distribution. In particular, when $p_\theta(x) = \mathcal{N}(x \mid 0, I)$,

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}[q_\phi(x \mid y) \parallel p_\theta(x)] \\
&= \frac{1}{2} \left[\mathrm{tr}(\Sigma_\phi(y)) + \|\mu_\phi(y)\|^2 - \log |\Sigma_\phi(y)| - n_x \right]. \qquad (7)
\end{aligned}$$

The ELBO can be maximized for the parameter(θ, ϕ) to train VAE.

III. Deterministic Nonlinear System Identification with Neural Networks

In this section, we briefly introduce nonlinear system identification for deterministic systems proposed in our work [5].

We consider the following deterministic discrete-time nonlinear dynamical system in this section:

$$x_{t+1} = f(x_t, u_t), \qquad y_t = h(x_t) \qquad (8)$$

with $t \in \mathbb{Z}$ the time index, $x_t \in \mathbb{R}^{n_x}$ the state, $y_t \in \mathbb{R}^{n_y}$ the output, $u_t \in \mathbb{R}^{n_u}$ the input, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ the state-transition map and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ the output map. As in the usual setting for system identification, x_t is a latent variable. In the following discussion, the fundamental properties of the proposed method for nonlinear target systems are presented in the absence of measurement noise and disturbances.

For convenience, we define the sequence of input and output signals in the past and future from a certain time t as follows:

$$\begin{aligned}
u_t^{\mathrm{p}} &:= \begin{bmatrix} u_{t-\mathrm{h_p}} \\ u_{t-\mathrm{h_p}+1} \\ \vdots \\ u_{t-1} \end{bmatrix}, \quad y_t^{\mathrm{p}} := \begin{bmatrix} y_{t-\mathrm{h_p}} \\ y_{t-\mathrm{h_p}+1} \\ \vdots \\ y_{t-1} \end{bmatrix}, \\
u_t^{\mathrm{f}} &:= \begin{bmatrix} u_t \\ u_{t+1} \\ \vdots \\ u_{t+\mathrm{h_f}-1} \end{bmatrix}, \quad y_t^{\mathrm{f}} := \begin{bmatrix} y_t \\ y_{t+1} \\ \vdots \\ y_{t+\mathrm{h_f}-1} \end{bmatrix}
\end{aligned} \qquad (9)$$

where the superscripts p and f indicate past and future, and $\mathrm{h_p} \in \mathbb{N}$ and $\mathrm{h_f} \in \mathbb{N}$ are horizons for corresponding directions, respectively. Using these notations, we formulate the problem as follows.

Problem 1. Given the measured input-output data $\{(u_t, y_t)\}_{t=-\mathrm{h_p}+1}^{T+\mathrm{h_f}-1}$, and the design parameters $\mathrm{h_p}, \mathrm{h_f}, n_{\hat{x}} \in \mathbb{N}$, construct a model that consists of a state estimator E_ϕ, which maps $(u_t^{\mathrm{p}}, y_t^{\mathrm{p}}) \mapsto \hat{x}_t$, and a predictor P_θ, which maps $(\hat{x}_t, u_t^{\mathrm{f}}) \mapsto y_t^{\mathrm{f}}$. Here, $\hat{x}_t \in \mathbb{R}^{n_{\hat{x}}}$ is a state equivalent to x_t but in an arbitrary coordinate system and $T \in \mathbb{N}$ indicates the size of the dataset.

Remark 2. In the usual state-space system identification, the goal is to construct a model of f and h, i.e., a state-space model. However, in many applications, including model predictive control, the state estimator and multi-step ahead predictor are often reconstructed from the resulting state-space model. In particular, for nonlinear systems that are difficult to analyze, the merits of going through state-space models are questionable. Therefore, we focus here on the construction of the state estimator E_ϕ and predictor P_θ. Once the state estimator is obtained, however, it is easy to estimate the state and construct the state-space model.

Then, we make assumptions to guarantee the existence of the solution of Problem 1.

Definition 3 (uniform k-observability)**.** If the mapping

$$\mathbb{R}^{n_x} \times (\mathbb{R}^{n_u})^k \to (\mathbb{R}^{n_y})^k \times (\mathbb{R}^{n_u})^k \quad \text{by} \quad (x,u) \mapsto \Big(h^k(x,u), u\Big) \tag{10}$$

is injective, the system (8) is said to be uniformly k-observable [14].

Assumption 4. The system (8) is uniformly k-observable, where $k = \min\left(\mathrm{h_p}, \mathrm{h_f}\right)$.

Remark 5. According to [15], typical dynamical systems are k-observable for $k \geq 2n_x + 1$.

Assumption 6. The dimension of the state estimate is large enough, that is, $n_{\hat{x}} \geq n_x$.

In this method, we train the neural network which has the structure of Fig. 2, where the state estimator $E_\phi : (\mathbb{R}^{n_u})^{\mathrm{h_p}} \times (\mathbb{R}^{n_y})^{\mathrm{h_p}} \to \mathbb{R}^{n_{\hat{x}}}$ maps the past input and output to the current state, and the predictor $P_\theta : \mathbb{R}^{n_{\hat{x}}} \times (\mathbb{R}^{n_u})^{\mathrm{h_f}} \to (\mathbb{R}^{n_y})^{\mathrm{h_f}}$ maps the current state and the future input to the future output, and θ and ϕ are parameters of the respective neural networks.

We set $n_{\hat{x}} < \mathrm{h_p} n_y$ for the bottleneck structure of the network to take a lower-dimensional state, and the training problem is formulated as the following minimization problem

$$\min_{\theta,\phi} \quad \frac{1}{T}\sum_{t=1}^{T} \Big\| y_t^{\mathrm{f}} - \underbrace{P_\phi\big(\underbrace{E_\theta\left(u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)}_{\text{estimate of } x_t}, u_t^{\mathrm{f}}\big)}_{\text{prediction of } y_t^{\mathrm{f}}} \Big\|^2. \tag{11}$$

Since the state x_t cannot be measured, the state estimator E_ϕ and predictor P_θ are coupled in series and trained in the same way as the encoder and decoder in autoencoder. To solve (11), we can utilize modern batch optimization algorithms for deep learning (e.g. Adam [16]). This method can be interpreted as nonlinear extension of classical subspace identification for linear systems. For more detail, see [5].

IV. Nonlinear System Identification with Uncertainty

The method presented in the previous section treats the target system as a deterministic entity and does not account for stochastic uncertainties arising from noise and modeling errors. In this paper, we introduce an identification approach that considers uncertainties by incorporating the principles of VAE.

In this section, we consider stochastic discrete-time nonlinear system

$$x_{t+1} \sim p(x_{t+1}|x_t, u_t), \qquad y_t \sim p(y_t|x_t) \tag{12}$$

and introduce an assumption.

Assumption 7. The system is not embedded in a closed-loop. That is, future inputs are independent of the current state $p(u_\tau \,|\, x_t) = p(u_\tau)$ for $\tau \geq t$.

Let's modify the state estimator and the output predictor so that they outputs an approximate posterior distribution $q_\phi^{\hat{x}}(\hat{x}_t|u_t^{\mathrm{p}}, y_t^{\mathrm{p}})$ and a predictive distribution of future output sequence $p_\theta^{y^{\mathrm{f}}}(y_t^{\mathrm{f}}|x_t, u_t^{\mathrm{f}})$, respectively, instead of a point estimate. Note that we introduce the approximated posterior (in this paper, we use a normal distribution) since the exact posterior distribution is generally complicated by nonlinearity. Restricting these distributions to normal distributions makes it possible to represent the distributions using only mean and covariance, and we can parameterize them as

$$q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right) = \mathcal{N}\Big(\hat{x}_t \,\Big|\, \mu_\phi^{\hat{x}}(u_t^{\mathrm{p}}, y_t^{\mathrm{p}}), \Sigma_\phi^{\hat{x}}(u_t^{\mathrm{p}}, y_t^{\mathrm{p}})\Big), \tag{13}$$

$$p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, x_t, u_t^{\mathrm{f}}\right) = \mathcal{N}\Big(y_t^{\mathrm{f}} \,\Big|\, \mu_\theta^{y^{\mathrm{f}}}(\hat{x}_t, u_t^{\mathrm{f}}), \Sigma_\theta^{y^{\mathrm{f}}}(\hat{x}_t, u_t^{\mathrm{f}})\Big). \tag{14}$$

by using neural networks

$$\begin{aligned}
\mu_\phi^{\hat{x}} &: (\mathbb{R}^{n_u})^{\mathrm{h_p}} \times (\mathbb{R}^{n_y})^{\mathrm{h_p}} \to \mathbb{R}^{n_{\hat{x}}}, \\
\Sigma_\phi^{\hat{x}} &: (\mathbb{R}^{n_u})^{\mathrm{h_p}} \times (\mathbb{R}^{n_y})^{\mathrm{h_p}} \to \mathbb{R}^{n_{\hat{x}} \times n_{\hat{x}}}, \\
\mu_\theta^{y^{\mathrm{f}}} &: \mathbb{R}^{n_{\hat{x}}} \times (\mathbb{R}^{n_u})^{\mathrm{h_f}} \to (\mathbb{R}^{n_y})^{\mathrm{h_f}}, \\
\Sigma_\theta^{y^{\mathrm{f}}} &: \mathbb{R}^{n_{\hat{x}}} \times (\mathbb{R}^{n_u})^{\mathrm{h_f}} \to \left(\mathbb{R}^{n_y \times n_y}\right)^{\mathrm{h_f} \times \mathrm{h_f}}
\end{aligned} \tag{15}$$

with parameters ϕ and θ.

Then, the problem can be stated as follows.

Problem 8. Given the measured input-output data $\{(u_t, y_t)\}_{t=-\mathrm{h_p}+1}^{T+\mathrm{h_f}-1}$, and the design parameters $\mathrm{h_p}, \mathrm{h_f}, n_{\hat{x}} \in \mathbb{N}$, construct neural networks (15) that model the distributions of current state $\hat{x}$ and future output y^{f} as in (13) and (14). Here, $\hat{x}_t \in \mathbb{R}^{n_{\hat{x}}}$ is a state equivalent to x_t but in an arbitrary coordinate system and $T \in \mathbb{N}$ indicates the size of the dataset.

To solve this problem, the ELBO of the marginal log likelihood $\log p_\theta^{y^{\mathrm{f}}}(y_t^{\mathrm{f}} \,|\, u_t^{\mathrm{f}})$ is calculated in the same way as the VAE, as

$$\log p\left(y_t^{\mathrm{f}} \,\middle|\, u_t^{\mathrm{f}}\right) = \log \int p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) p\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{f}}\right) d\hat{x}_t$$

(since x_t and u_t^{f} are independent from Assumption 7)

$$\begin{aligned}
&= \log \int p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) p(\hat{x}_t) d\hat{x}_t \\
&= \log \int \frac{p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) p(\hat{x}_t)}{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right) d\hat{x}_t \\
&= \log \mathbb{E}_{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} \left[\frac{p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) p(\hat{x}_t)}{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} \right] \\
&\geq \mathbb{E}_{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} \left[\log \frac{p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) p(\hat{x}_t)}{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} \right] \\
&= \mathbb{E}_{q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right)} \left[\log p_\theta^{y^{\mathrm{f}}}\left(y_t^{\mathrm{f}} \,\middle|\, \hat{x}_t, u_t^{\mathrm{f}}\right) \right] \\
&\qquad - \mathrm{D_{KL}}\left[q_\phi^{\hat{x}}\left(\hat{x}_t \,\middle|\, u_t^{\mathrm{p}}, y_t^{\mathrm{p}}\right) \parallel p(\hat{x}_t) \right] \\
&:= \mathcal{L}\left(u_t^{\mathrm{p}}, y_t^{\mathrm{p}}, u_t^{\mathrm{f}}, y_t^{\mathrm{f}}; \theta, \phi\right).
\end{aligned} \tag{16}$$

And the parameters ϕ and θ are designed by maximizing the ELBO (16) using stochastic gradient descent method to obtain $\mu_\phi^{\hat{x}}$ and $\Sigma_\phi^{\hat{x}}$, which constitute the state estimator, and $\mu_\theta^{y^{\mathrm{f}}}$ and $\Sigma_\theta^{y^{\mathrm{f}}}$, which constitute the output predictor. Similar to the VAE described in Section II, the stochastic gradient of the

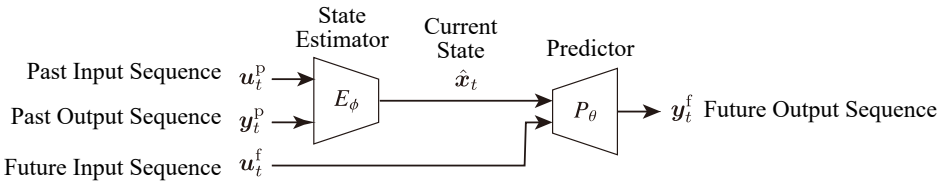


Fig. 2. Network structure. The sequences of past inputs u_t^{p} and outputs y_t^{p} are once compressed into state variables $\hat{x}_t$, and then reflected in the prediction of the sequence of future outputs y_t^{f}.

expectation term of the ELBO is calculated by a single sample Monte Carlo estimate.

The obtained state estimator and output predictor can be used to predict the mean and the variance of the state and future output (see Fig. 3). For example, the mean of the future output is evaluated as

$$\mathbb{E}_{q_\phi^{\hat{x}}(\hat{x}_t \mid u_t^{\mathrm{p}}, y_t^{\mathrm{p}})} \left[\mu_\theta^{y^{\mathrm{f}}}(\hat{x}_t, u_t^{\mathrm{f}}) \right], \tag{17}$$

and the variance of future output is evaluated as

$$\mathbb{E}_{q_\phi^{\hat{x}}(\hat{x}_t \mid u_t^{\mathrm{p}}, y_t^{\mathrm{p}})} \left[\Sigma_\theta^{y^{\mathrm{f}}}(\hat{x}_t, u_t^{\mathrm{f}}) \right]. \tag{18}$$

Of course, it is difficult to calculate the expectations analytically, so for practical use, it is necessary to approximate the expected value using the Monte Carlo method. In contract, the mean and variance of the state can be directly obtained by $\mu_\phi^{\hat{x}}(u_t^{\mathrm{p}}, y_t^{\mathrm{p}})$ and $\Sigma_\phi^{\hat{x}}(u_t^{\mathrm{p}}, y_t^{\mathrm{p}})$. Note, however, that there are degrees of freedom in how the coordinate system is taken in the state space, and it is not known what coordinate system will be chosen.

V. Numerical Examples

In this section, the proposed method is validated through numerical examples.

A. Linear System

First, a linear system is identified by the proposed method. Since the optimal state estimator for linear system is known to be Kalman filter, we can evaluate the state estimator obtained by the proposed method by comparing it with Kalman filter. It is also easy to evaluate the output predictor by computing the distribution of future output according to the distribution of states obtained by Kalman filter and the exact model.

The state-space equation of the target system is as follows:

$$x_{t+1} = \begin{bmatrix} 0.9 & 0.8 \\ 0 & 0.1 \end{bmatrix} x_t + \begin{bmatrix} -1 \\ 0.1 \end{bmatrix} u_t + w_t \tag{19}$$

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} x_t + v_t \tag{20}$$

where $w_t \sim \mathcal{N}(0, 0.5I)$ and $v_t \sim \mathcal{N}(0, 1)$. Here, we set hyperparameters as $\mathrm{h_p} = \mathrm{h_f} = 20$ and $n_{\hat{x}} = 2$. All $\mu_\phi^{\hat{x}}, \Sigma_\phi^{\hat{x}}, \mu_\theta^{y^{\mathrm{f}}}, \Sigma_\theta^{y^{\mathrm{f}}}$ are neural networks with 2 hidden layers, 64 neurons and Rectified Linear Unit (ReLU) activation function. The neural networks were trained using Adam [16]. In this experiment, we constructed the neural network so that the values of $\Sigma_\phi^{\hat{x}}$ and $\Sigma_\theta^{y^{\mathrm{f}}}$ are diagonal matrices, as often seen in VAE studies. The input for the system is generated by sampling u_t from $\mathcal{N}(0, 1)$. In addition to $T = 90\,000$ data for training, 10 000 data were prepared for validation, and optimization was performed until the ELBO calculated for the validation data did not improve for 100 successive iterations.

For testing the obtained model, we generated data for different random realizations. Fig. 4 show the mean and variance of the predicted output, calculated as in (17) and (18) at time t. In the figure, the blue crosses indicate the predicted distribution of the k-step ahead prediction $\left(y_t^{\mathrm{f}}\right)_k$ at time t. For comparison, the distribution of states at time t is estimated with the stationary Kalman filter, which is known to be optimal, and the distribution of y_t^{f} computed from this distribution using the exact model is shown by the black dashed line in the figure.

As seen in the figure, the state estimator and output predictor obtained by the proposed method in this example are in close agreement with the Kalman filter and the correct model, and the proposed approach is promising.

B. Nonlinear System

Next, to demonstrate the validity of the proposed method in nonlinear systems, we apply the proposed method to a nonlinear system based on the cascaded tanks example in [17]. The state-space equation of the system is

$$\begin{bmatrix} x_{t+1,1} \\ x_{t+1,2} \end{bmatrix} = \begin{bmatrix} \max\left(x_{t,1} - k_1\sqrt{x_{t,1}} + k_2\left(u_t + \sqrt{0.1x_{t,1}}w_t\right),\ 0\right) \\ \max\left(x_{t,2} + k_3\sqrt{x_{t,1}} - k_4\sqrt{x_{t,2}},\ 0\right) \end{bmatrix}, \tag{21}$$

$$y_t = x_{t,2} + v_t. \tag{22}$$

where $k_1 = 0.5$, $k_2 = 0.4$, $k_3 = 0.2$, $k_4 = 0.3$; $x_{t,1}, x_{t,2}$ are the water levels in the upper and lower tanks, respectively; u_t represents the pump voltage; and $w_t \sim \mathcal{N}\left(0, 2^2\right)$ and $v_t \sim \mathcal{N}\left(0, 0.1^2\right)$ are process noise and measurement noise, respectively. Note in particular that the process noise w_t is scaled by $\sqrt{0.1x_{t,1}}$ to add dynamic variation to the distribution. The hyperparameters are set as $\mathrm{h_p} = \mathrm{h_f} = 20$ and $n_{\hat{x}} = 2$. These are the same as in the example in Section V-A, and also let the neural networks $\mu_\phi^{\hat{x}}, \Sigma_\phi^{\hat{x}}, \mu_\theta^{y^{\mathrm{f}}}, \Sigma_\theta^{y^{\mathrm{f}}}$ to have the same structure. The inputs to the system for generating training data are as follows:

$$u_t = \min\left(\max\left(u_t^0, 0\right), 5\right), \qquad u_t^0 \sim \mathcal{N}(2, 5^2), \tag{23}$$

The size of the dataset for training is 900 000. 100 000 data points were prepared for validation, and optimization was performed until the ELBO calculated for the validation data did not improve for 100 successive iterations.

For testing the obtained model, u_t is generated in the same way as for the training data. Since an exact solution cannot

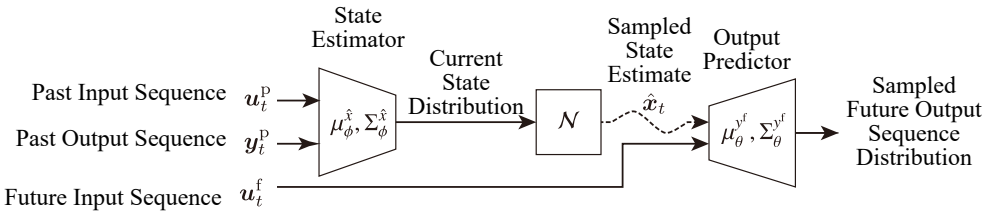
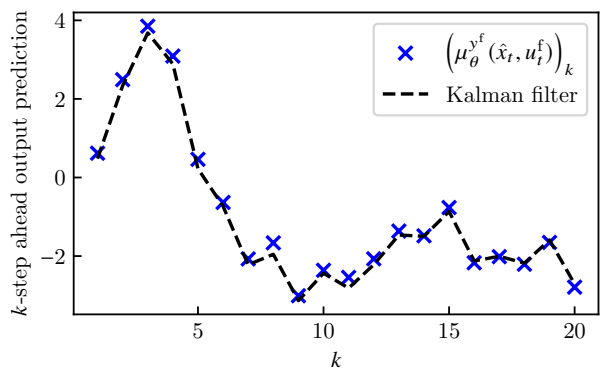
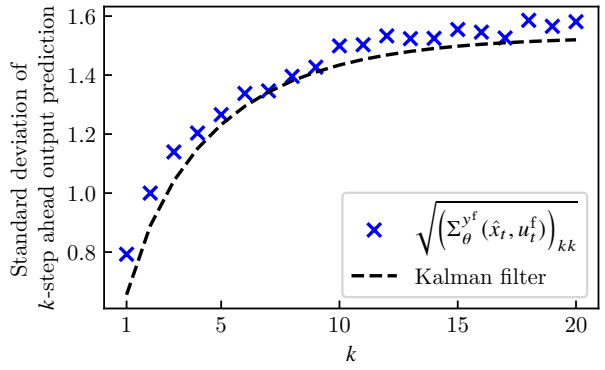


Fig. 3. Diagram of the model constructed by the proposed method. The distribution of states and future outputs are estimated instead of point estimates.



(a) Mean



(b) Variance

Fig. 4. Estimated distribution of future output

be obtained in the case of nonlinear systems, the validity of the model is confirmed by comparing the distribution of the prediction by the model and the output from the system. More specifically, u_t^{p}, y_t^{p}, and u_t^{f} at a certain time t were input to the model, and the operation of sampling a single $\hat{x}_t$ from (13) and $\hat{y}_t^{\mathrm{f}}$ from (14) was performed 1000 times, changing the random realization in sampling $\hat{x}_t$ and $\hat{y}_t^{\mathrm{f}}$ and the noise in y_t^{p}, while the input realization u_t^{p} and u_t^{f} is unchanged. The distribution of obtained y_t^{f} and $\hat{y}_t^{\mathrm{f}}$ are summarized in Fig. 5. Since the target system is nonlinear, the results at four different operating points (for various random realizations of the input) are summarized in four subfigures. In the figure, the blue curve and the orange band show the median and the upper and lower quartiles (25% – 75%) of the prediction from the model $\hat{y}_t^{\mathrm{f}}$. On the other hand, the box plots in the figure show the distribution of the output from the system y_t^{f}.

As can be seen in the figure, the quartiles and medians are in general agreement, indicating that the proposed method provides a reasonable model for the dynamics and uncertainty of the nonlinear target system, including the changes in variance that depend on the operating point. Because the proposed method models the states and outputs with a normal distribution, the deviation tends to be larger in regions where there are many outliers and the actual distribution is considered to be far from the normal distribution (the upper two figures).

VI. Conclusion

In this paper, we proposed a nonlinear system identification method for constructing models that estimate the distribution of states and outputs. The method is based on a nonlinear subspace identification method using a neural network with a bottleneck structure and applies the concept of a variational autoencoder. The validity of the proposed method was verified through numerical examples by comparing the obtained model with the optimal Kalman filter for a linear system and by comparing the distribution of the model output with the actual output for a nonlinear system.

In future work, we plan to verify this approach in combination with control approaches that can utilize stochastic information, such as probabilistically constrained model predictive controller [6] and scenario-based approach [7].

References

[1] Lennart Ljung. *System identification: Theory for the user*. Prentice Hall, 2 edition, 1999.

[2] Alexandre Mauroy and Jorge Goncalves. Parameter estimation and identification of nonlinear systems with the koopman operator. In Alexandre Mauroy, Igor Mezić, and Yoshihiko Susuki, editors, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, pages 335–357. Springer International Publishing, Cham, 2020.

[3] Lennart Ljung, Carl Andersson, Koen Tiels, and Thomas B. Schön. Deep learning and system identification. *IFAC-PapersOnLine*, 53(2):1175–1181, 2020. 21st IFAC World Congress.

[4] Tohru Katayama. *Subspace methods for system identification*. Springer, 1 edition, 2005.

[5] Keito Yamada, Ichiro Maruta, and Kenji Fujimoto. Subspace state-space identification of nonlinear dynamical system using deep neural network with a bottleneck. In *Proc. of 12th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, pages 102–107, 2023.

[6] Pu Li, Moritz Wendt, and Günter Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38(7):1171–1176, 2002.

[7] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

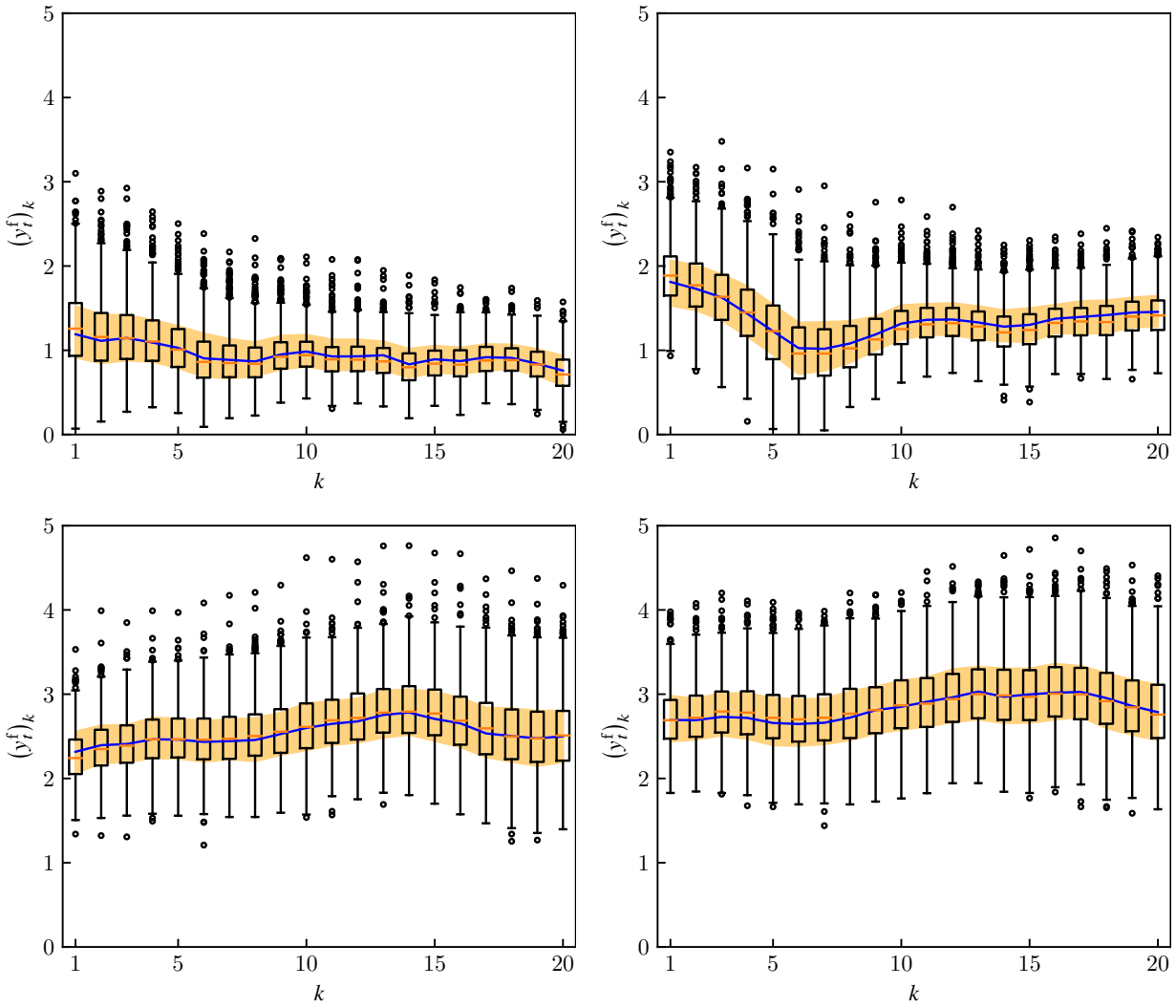


Fig. 5. Comparison between the distribution of the system output (box plots) and the prediction by the model (blue line with orange band)

[9] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[10] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[11] Daniel Gedon, Niklas Wahlström, Thomas B. Schön, and Lennart Ljung. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, 54(7):481–486, 2021. 19th IFAC Symposium on System Identification SYSID 2021.

[12] Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models. *Advances in Neural Information Processing Systems*, 34:10234–10245, 2021.

[13] S. Joe Qin and Thomas A. Badgwell. An overview of nonlinear model predictive control applications. In Frank Allgöwer and Alex Zheng, editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 369–392. Birkhäuser Basel, 2000.

[14] P.E. Moraal and J.W. Grizzle. Observer design for nonlinear systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 40(3):395–404, 1995.

[15] J. Stark, D.S. Broomhead, M.E. Davies, and J. Huke. Takens embedding theorems for forced and stochastic systems. *Nonlinear Analysis: Theory, Methods & Applications*, 30(8):5303–5314, 1997. Proceedings of the Second World Congress of Nonlinear Analysts.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Torbjörn Wigren and Johan Schoukens. Three free data sets for development and benchmarking in nonlinear system identification. In *Proc. of 2013 European Control Conference (ECC)*, pages 2933–2938. IEEE, 2013.