Replica Exchange Particle-Gibbs Method with Ancestor Sampling

Inoue, Hiroaki
Hukushima, Koji
Omori, Toshiaki

(Citation)
Journal of the Physical Society of Japan,89(10):104801-104801

(Issue Date)
2020-10-15

(Resource Type)
journal article

(Version)
Version of Record

Journal of the Physical Society of Japan **89**, 104801 (2020)
https://doi.org/10.7566/JPSJ.89.104801

Replica Exchange Particle-Gibbs Method with Ancestor Sampling

Hiroaki Inoue[1], Koji Hukushima[2,3], and Toshiaki Omori[1,4*]

[1]*Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University, Kobe 657-8501, Japan*
[2]*Graduate School of Arts and Sciences, The University of Tokyo, Meguro, Tokyo 153-8902, Japan*
[3]*Komaba Institute for Science, The University of Tokyo, Meguro, Tokyo 153-8902, Japan*
[4]*Organization for Advanced and Integrated Research, Kobe University, Kobe 657-8501, Japan*

(Received June 4, 2020; accepted July 27, 2020; published online September 3, 2020)

Particle-Gibbs (PG) method is a Markov chain Monte Carlo (MCMC) method for sampling from the joint posterior distribution of latent variables and parameters in a nonlinear state space model, which is a probabilistic time-series model. Recently, particle-Gibbs with ancestor sampling (PGAS) was proposed as a method with better sampling efficiency of PG. However, even with PG and PGAS, it is often difficult to realize sampling from the joint posterior distribution with a finite number of samples. In this paper, we propose replica exchange particle-Gibbs with ancestor sampling (REPGAS) as a method to overcome this problem by combining PGAS with the replica exchange method. We also demonstrate the effectiveness of the proposed method by using simulated data obtained from a benchmark nonlinear state space model and the Izhikevich model, which is a computational model of the membrane potential of a neuron.

1. Introduction

In recent years, time-series data have been investigated extensively by using state space models.[1–18] State space models are probabilistic models that assume the existence of latent variables. The latent variables cannot be observed directly against the background from which observations are obtained. In the state space models, we assume a system model describing dynamical behavior of latent variables and an observation model describing a process that relates latent variables to observation values.

State space models have been utilized in various fields (e.g., physics, earth science and brain science) in order to estimate latent variables[9,15,17,19] and forecast observation values.[6,16,18] When the values of parameters in the state space model are known, latent variables are estimated using sequential Bayesian filter such as Kalman filter and particle filter, and predictive observation values can be obtained. If the values of the model parameters are unknown, however, it is necessary to estimate the parameters and latent variables in the state space model simultaneously.

A method combining a sequential Bayesian filter and expectation–maximization (EM) algorithm has been employed to estimate latent variables and parameters of state space models.[2,7,9,15,20,21] Although the method is an iterative method for point estimation of parameters and its convergence to a local optimum is guaranteed, its dependence on the initial value makes it hard for it to estimate a global optimum in a complex state space model.

In recent years, a method called particle-Gibbs (PG) was proposed for estimating latent variable and parameter distributions simultaneously.[8,11,13,14] The PG is the method combining sequential Monte Carlo (SMC) methods (also known as particle filter),[3,4,7–9,11–15] a method for approximating the latent variable distributions as a set of particles, with Gibbs sampling that is a type of Markov chain Monte Carlo (MCMC) method.[7,22–24] Using PG, it is possible to estimate the joint posterior distribution of latent variables and parameters in state space models. In addition, particle-Gibbs with ancestor sampling (PGAS) was proposed as a method for improving the sampling efficiency of PG.[11,13,14] In the SMC method used in the PG, the particles are resampled at each time step, so that there is a problem that only a few initial particles survive in the end and degenerate.[8,11,13,14] In PGAS, in order to avoid this degeneracy of latent variables in the SMC method, an ancestor sampling step is introduced; one of particles at each time is sampled from particles in the previous time as the ancestor of conditioned sample, and the same effect as performing backward simulation is obtained by performing forward simulation based on conditional SMC with ancestor sampling[11,13,14] in PG. However, when the target distribution is a multimodal distribution and the distance between peaks is large, it is difficult to sample from the target distribution with a finite number of samples, and the estimated distribution is strongly influenced by the initial values.

In this paper, we propose a replica exchange particle-Gibbs with ancestor sampling (REPGAS), which combines PGAS with the replica exchange method[25,26] in order to overcome such problems of PGAS. The replica exchange method is a type of MCMC method, that exchanges configurations between different temperatures and improves the MCMC framework. We show that the proposed method, REPGAS, can overcome the problem of initial value dependence of PGAS by estimating distributions of latent variables and parameters. We show this by using simulated data from a benchmark nonlinear state space model.[1,8,12–14] Furthermore, we show that it is possible to obtain samples more efficiently than in the conventional method in a realistic model by applying the proposed method to Izhikevich neuron model, which is a computational model of neuronal membrane potential expressed by multi-dimensional differential equations.[27,28]

2. Method

In this section, we first describe state space models using probabilistic models. Next, we explain PGAS, a conventional method for estimating the parameters and latent variables in a state space model simultaneously. After that, we propose REPGAS combining PGAS with the replica exchange method for overcoming the problem of initial value dependence.

2.1 *State space models*

A state space model can be represented by the graphical

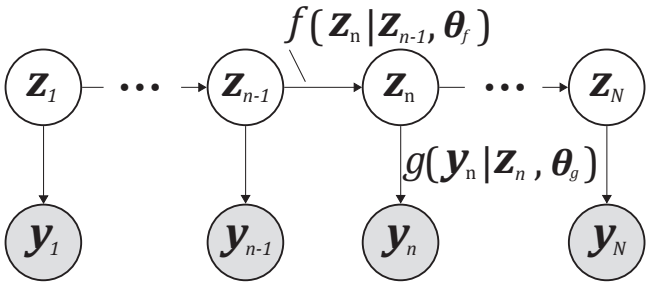

Fig. 1. Graphical model of a state space model. $z_{1:N}$ and $y_{1:N}$ are latent variables and observations, respectively. The arrow from z_{n-1} to z_n represents a system model [Eq. (1)], and the arrow from z_n to y_n represents an observation model [Eq. (2)]. The goal of this paper is to estimate the distribution of parameters $\Theta = \{\theta_f, \theta_g\}$ and latent variables $z_{1:N}$ from observations $y_{1:N}$.

model shown in Fig. 1, and there are latent variables $z_{1:N} = \{z_1, z_2, \ldots, z_N\}$ which cannot be observed directly in the background in the time range $\{1, 2, \ldots, N\}$, and observation values $y_{1:N} = \{y_1, y_2, \ldots, y_N\}$. At time n the state space model is expressed as follows:

$$z_n \sim f(z_n \mid z_{n-1}, \theta_f), \tag{1}$$

$$y_n \sim g(y_n \mid z_n, \theta_g), \tag{2}$$

where $f(z_n \mid z_{n-1}, \theta_f)$ is called the system model, and $g(y_n \mid z_n, \theta_g)$ is called the observation model. Here, θ_f and θ_g are the parameters in the system model and observation model, respectively. The system model represents the process of updating latent variables, and the observation model represents the process of obtaining observations from the latent variables.

If the model of interest is linear Gaussian state space model, the system model and observation model are expressed as follows:

$$f(z_n \mid z_{n-1}, \theta_f) = \mathcal{N}(z_n \mid \boldsymbol{A} z_{n-1}, \boldsymbol{\Sigma}_f), \tag{3}$$

$$g(y_n \mid z_n, \theta_g) = \mathcal{N}(y_n \mid \boldsymbol{B} z_n, \boldsymbol{\Sigma}_g), \tag{4}$$

where $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Also, $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{B}$ is the observation matrix, and $\boldsymbol{\Sigma}_f$ and $\boldsymbol{\Sigma}_g$ are the covariance matrices. In this case, the parameters are $\theta_f = \{\boldsymbol{A}, \boldsymbol{\Sigma}_f\}$ and $\theta_g = \{\boldsymbol{B}, \boldsymbol{\Sigma}_g\}$.

In the linear Gaussian state space model, it is possible to estimate the latent variables $z_{1:N}$ employing the Kalman filter. However, in the nonlinear or non-Gaussian state space model, it is necessary to employ the SMC methods.

2.2 *Particle-Gibbs with ancestor sampling*

We explain a conventional method for simultaneously estimating the parameters and latent variables in a state space model simultaneously. PGAS is a method for improving the sampling efficiency of PG. It is an MCMC method combining the SMC method with Gibbs sampling.[8,11,13,14] In PGAS, the values of parameters Θ and latent variables $z_{1:N}$ are initialized as $\Theta[0] = \{\theta_f[0], \theta_g[0]\}$ and $z_{1:N}[0]$, and the samples of latent variables and parameters are obtained alternately.

The k-th sample of latent variables $z_{1:N}[k]$ is obtained from $p(z_{1:N} \mid z_{1:N}[k-1], y_{1:N}, \Theta[k-1])$ with the conditional SMC with ancestor sampling given the previous sample of latent variables $z_{1:N}[k-1]$ as conditioned sample and the previous sample of parameters $\Theta[k-1]$. In the conditional SMC with ancestor sampling, the distribution of latent variables is approximated by particles $\{z_{1:N}^{(1)}, z_{1:N}^{(2)}, \ldots, z_{1:N}^{(M)}\}$ as follows:

$$p(z_{1:N} \mid z_{1:N}[k-1], y_{1:N}, \Theta[k-1]) \simeq \frac{1}{M} \sum_{i=1}^{M} \delta(z_{1:N} - z_{1:N}^{(i)}), \tag{5}$$

where $z_{1:N}^{(i)}$ is i-th particle, M is the number of particles, and $\delta(z_{1:N})$ is the Dirac delta distribution. To obtain particles, at the time step n, the indices of ancestor particles at previous time step $n-1$ $\{A_{n-1}^1, A_{n-1}^2, \ldots, A_{n-1}^{M-1}\}$ are sampled based on the normalized weights $\{W_{n-1}^1, W_{n-1}^2, \ldots, W_{n-1}^M\}$ obtained as follows:

$$W_{n-1}^i = \frac{w_{n-1}^i}{\sum_{j=1}^{M} w_{n-1}^j}, \tag{6}$$

$$w_{n-1}^i = g(y_{n-1} \mid z_{n-1}^{(i)}, \theta_g). \tag{7}$$

Latent variables z_n at the time step n are sampled from the system model $f(z_n \mid z_{n-1}^{(A_{n-1}^i)}, \theta_f)$. The particles are set to be $z_{1:n}^{(i)} \leftarrow \{z_{1:n-1}^{(A_{n-1}^i)}, z_n^{(i)}\}$ for particle numbers $i \in \{1, 2, \ldots, M-1\}$, while the M-th particle is set to be the previous sample $z_{1:n}^{(M)} \leftarrow \{z_{1:n-1}^{(A_{n-1}^M)}, z_n[k-1]\}$. Here, the index of ancestor particle A_{n-1}^M is sampled based on the weights $\{\hat{W}_{n-1}^1, \hat{W}_{n-1}^2, \ldots, \hat{W}_{n-1}^M\}$ calculated as follows:

$$\hat{W}_{n-1}^i = \frac{\hat{w}_{n-1}^i}{\sum_{j=1}^{M} \hat{w}_{n-1}^j}, \tag{8}$$

$$\hat{w}_{n-1}^i = W_{n-1}^i f(z_n[k-1] \mid z_{n-1}^{(i)}, \theta_f). \tag{9}$$

In PGAS, we iterate the above flow from time step 1 to N and the k-th sample of latent variables $z_{1:N}[k]$ is sampled based on weights $\{W_N^1, W_N^2, \cdots W_N^M\}$.

The k-th sample of parameters $\Theta[k]$ is obtained from $p(\Theta \mid z_{1:N}[k], y_{1:N})$ with an MCMC method (e.g., the Metropolis algorithm) given the k-th sample of latent variables $z_{1:N}[k]$.

PGAS iterating infinitely is guaranteed to sample from the target distribution. However, if the target distribution is multimodal and the peaks are apart, it is difficult to sample from the target distribution with a finite number of samples and the estimated distribution has strongly dependent on the initial values.

2.3 *Replica exchange particle-Gibbs with ancestor sampling*

In our study, we propose REPGAS combining PGAS and the replica exchange method for improving the problem influenced by the initial value of PGAS. Figure 2 presents the schematic diagram of REPGAS. The estimated distributions of the latent variables $z_{1:N}$ and parameter θ in the state space model are obtained by providing the time series data $y_{1:N}$ to REPGAS as inputs. In REPGAS, we introduce the extension variables called temperature into PGAS. By running PGAS in multiple temperatures and exchanging samples between temperatures as shown in the middle part of the figure, the

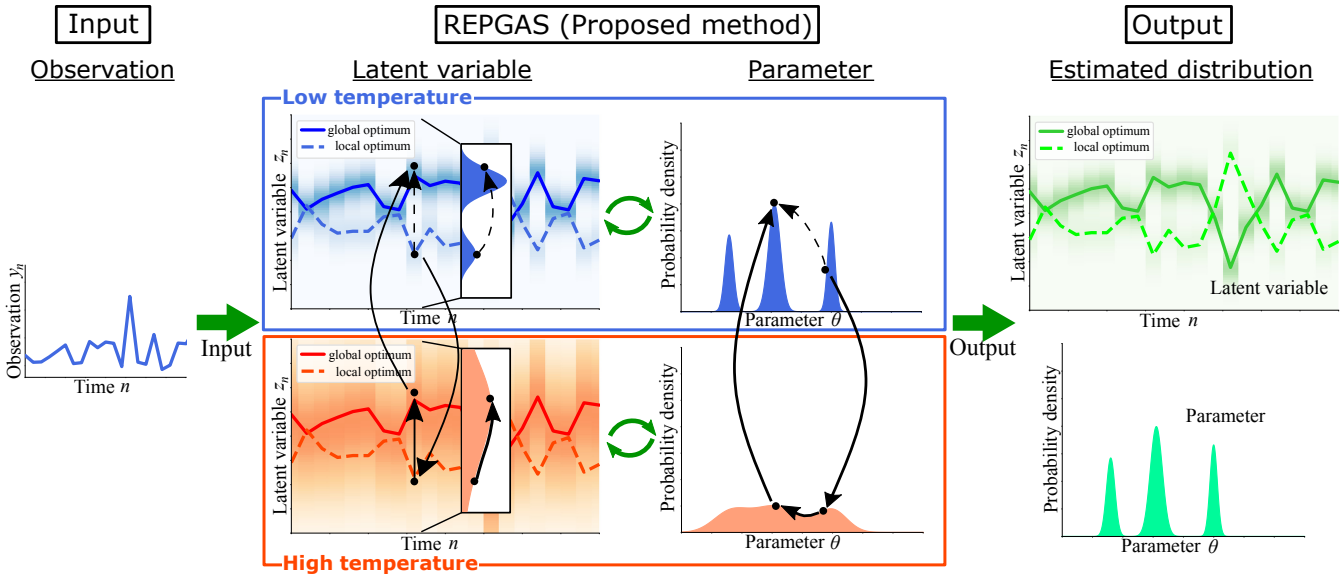

Fig. 2. (Color online) Schematic diagrams of REPGAS. (Left) Observations of time series data $y_{1:N}$ are given as the input. (Middle) The left figures are the distributions of latent variables and the right figures are the distributions of parameters in low and high temperatures. The arrows express the transitions of samples in REPGAS. In REPGAS, the transitions that are difficult to achieve with PGAS (dashed arrows) can be realized by passing through high temperature (solid arrows). (Right) The estimated distributions of latent variables $z_{1:N}$ and parameter θ are obtained as the output by collecting samples.

transitions that are difficult to achieve with PGAS can be realized by passing through high temperatures. Although Fig. 2 shows the case where both the latent variable z_n and the parameter θ are one-dimensional, REPGAS is also applicable to the case of multidimensional data.

We introduce temperatures $\boldsymbol{T} = [T^1, T^2, \ldots, T^R]$ (R is the number of temperatures) as extension variables and consider the following extended joint posterior distribution:

$$\pi_{\mathrm{EX}}(\{z_{1:N}\}, \{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}) = \frac{1}{Z_{\mathrm{EX}}} \prod_{r=1}^{R} \pi_{T^r}(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N}), \quad (10)$$

where Z_{EX} is the normalizing constant, and $\{z_{1:N}\}$ and $\{\boldsymbol{\Theta}\}$ are represented as $\{z_{1:N}\} = \{z^1_{1:N}, z^2_{1:N}, \ldots, z^R_{1:N}\}$ and $\{\boldsymbol{\Theta}\} = \{\boldsymbol{\Theta}^1, \boldsymbol{\Theta}^2, \ldots, \boldsymbol{\Theta}^R\}$, respectively. Furthermore, the joint posterior distribution at each temperature $\pi_{T^r}(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N})$ is represented as follows:

$$\pi_{T^r}(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N}) = \frac{1}{Z(T^r)} p(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N})^{\frac{1}{T^r}}, \quad (11)$$

where $Z(T^r)$ represents a partition function. At sufficiently high temperatures, the latent variables and the parameters almost follow a uniform distribution, independent of observed $\boldsymbol{y}_{1:N}$. The distribution with $T_1 = 1.0$ corresponds to the original posterior distribution to be investigated. The posterior distribution $p(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N})$ is obtained using Bayes' theorem as follows:

$$p(z^r_{1:N}, \boldsymbol{\Theta}^r \mid \boldsymbol{y}_{1:N}) = \frac{p(\boldsymbol{y}_{1:N} \mid z^r_{1:N}, \boldsymbol{\Theta}^r) p(z^r_{1:N} \mid \boldsymbol{\Theta}^r) p(\boldsymbol{\Theta}^r)}{p(\boldsymbol{y}_{1:N})}, \quad (12)$$

where $p(\boldsymbol{y}_{1:N})$ is a constant, and $p(\boldsymbol{\Theta})$ is a prior distribution of parameters $\boldsymbol{\Theta}$.

In REPGAS, we obtain samples of the latent variables $z^r_{1:N}$ and parameters $\boldsymbol{\Theta}^r$ for each temperature with PGAS according to Eq. (11), and exchange samples between temperatures T^r and T^{r+1} according to the following exchange probability:

$$p_{\mathrm{EX}} = \min(1, R_{\mathrm{EX}}), \quad (13)$$

$$R_{\mathrm{EX}} = \frac{\pi_{\mathrm{EX}}(\{z^*_{1:N}\}, \{\boldsymbol{\Theta}^*\} \mid \boldsymbol{y}_{1:N})}{\pi_{\mathrm{EX}}(\{z_{1:N}\}, \{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N})}, \quad (14)$$

where $\{z^*_{1:N}\}$ and $\{\boldsymbol{\Theta}^*\}$ are expressed as follows:

$$\{z^*_{1:N}\} = \{z^1_{1:N}, \ldots, z^{r+1}_{1:N}, z^r_{1:N}, \ldots, z^R_{1:N}\}, \quad (15)$$

$$\{\boldsymbol{\Theta}^*\} = \{\boldsymbol{\Theta}^1, \ldots, \boldsymbol{\Theta}^{r+1}, \boldsymbol{\Theta}^r, \ldots, \boldsymbol{\Theta}^R\}. \quad (16)$$

In REPGAS, it becomes possible to overcome the problem of initial value dependence in PGAS by passing through a high temperature state in the replica exchange method. Moreover, it also becomes possible to prevent increasing the calculation time because each PGAS is able to be run in parallel.

3. Results

In this section, it is shown by employing our REPGAS for the benchmark nonlinear state space model, that the joint posterior distribution of latent variables and parameters can be estimated from observations, that sampling efficiency is improved compared to PGAS, and that it is possible to overcome the problem of initial value dependence in PGAS. Furthermore, we also show that the efficiency of REPGAS is confirmed by applying it to the Izhikevich neuron model.

3.1 Benchmark nonlinear state space model

In this paper, to verify the effectiveness of the proposed method, we use the following benchmark nonlinear state space model:[1,8,12–14]

$$z_n \sim \mathcal{N}\left(z_n \,\middle|\, \frac{z_{n-1}}{a} + b\frac{z_{n-1}}{1+z^2_{n-1}} + c\cos(dn), \sigma^2_z\right), \quad (17)$$

$$y_n \sim \mathcal{N}\left(y_n \,\middle|\, \frac{z^2_n}{e}, \sigma^2_y\right), \quad (18)$$

where a, b, c, d, and e are constants, and σ_z and σ_y are standard deviations of the system model and observation model, respectively. In this model, the latent variables have

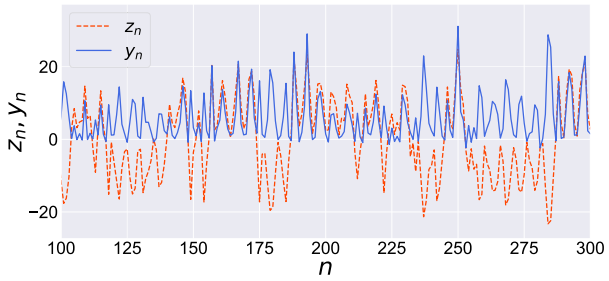

Fig. 3. (Color online) Observations y_n (blue solid line) and true latent variables z_n (red dashed line) obtained from the benchmark nonlinear state space model at time step $n = 100$–300.

both positive and negative values, but the observations are converted to only positive values through the observation model.

In the following numerical experiments, we estimate the joint posterior distribution of latent variables and parameters $p(z_{1:N}, \boldsymbol{\Theta} \mid y_{1:N})$ from the observations $y_{1:N}$ with the parameters $\{a, b, c, d, e, \sigma_z^2, \sigma_y^2\} = \{2, 25, 8, 1.2, 20, 10, 1\}$ and the number of data $N = 1500$. Here we focus on estimating the system model parameters $\boldsymbol{\Theta} = [a, b, c, d]$.

We show a portion of the data used in the following experiments in Fig. 3, where the vertical axis represents the value of observation y_n and latent variable z_n, the horizontal axis represents the time step n, the solid line is observation y_n, and the dashed line is latent variable z_n.

3.1.1 *Experiment to compare sampling efficiency*

To compare the sampling efficiency of PGAS and REPGAS, we estimate the joint posterior distribution of latent variables and parameters $p(z_{1:N}, \boldsymbol{\Theta} \mid y_{1:N})$ when the initial values of parameters $\boldsymbol{\Theta}[0]$ are true values. Here, the number of samples is $K = 10^6$, the number of burn-in samples is $K_{\text{burn-in}} = 5 \times 10^5$, and the number of particles is $M = 50$. The initial values of latent variables are $z_{1:N}[0] = [0, 0, \ldots, 0]$, and the number of replicas of REPGAS is $R = 90$.

We show in Fig. 4 the autocorrelation function results calculated with PGAS and REPGAS samples of parameters $\boldsymbol{\Theta}$ at $T^1 = 1.0$. In all the graphs, the vertical axis represents the value of the autocorrelation function and the horizontal axis represents the lag length of the autocorrelation function. The dashed lines are the results calculated with PGAS samples of parameters a, b, c, and d. The solid lines are the results calculated with REPGAS samples.

In all of the parameters a, b, c, and d, the decay of the autocorrelation is faster with REPGAS than it is with PGAS. Therefore, it was shown that the sampling efficiency of REPGAS is higher than of PGAS.

3.1.2 *Experiment to compare dependence on initial values*

To verify whether the proposed method, REPGAS, improves PGAS in terms of initial values dependence, we estimate the joint posterior distribution of latent variables and parameters $p(z_{1:N}, \boldsymbol{\Theta} \mid y_{1:N})$ when the initial values of parameters are $\boldsymbol{\Theta}[0] = [a, b, c, d] = [1.5, 28, 7, 1.195]$, which are far from the true values $[a, b, c, d] = [2, 25, 8, 1.2]$. Here, the number of samples is $K = 2 \times 10^6$, the number of burn-in samples is $K_{\text{burn-in}} = 10^6$, and the number of particles

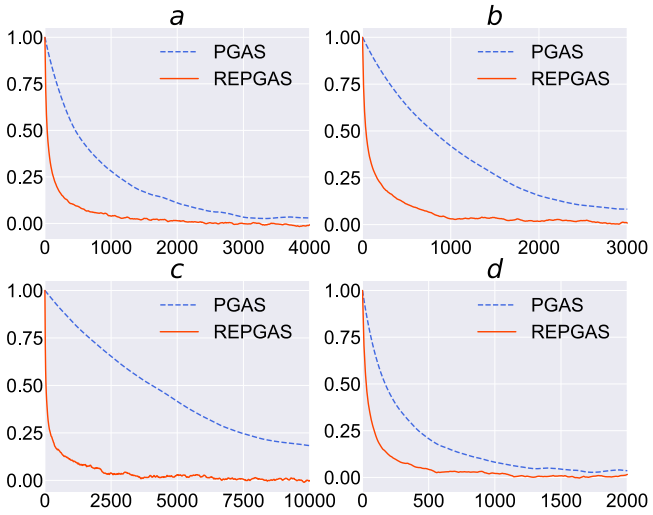

Fig. 4. (Color online) Autocorrelation function results of samples obtained by employing PGAS and REPGAS in the benchmark nonlinear state space model. In each graph, the vertical axis represents the value of the autocorrelation function and the horizontal axis represents the lag length of the autocorrelation function. The blue dashed lines are the results calculated with PGAS samples of parameters a, b, c, and d, the red solid lines are the results calculated with REPGAS samples at $T^1 = 1.0$.

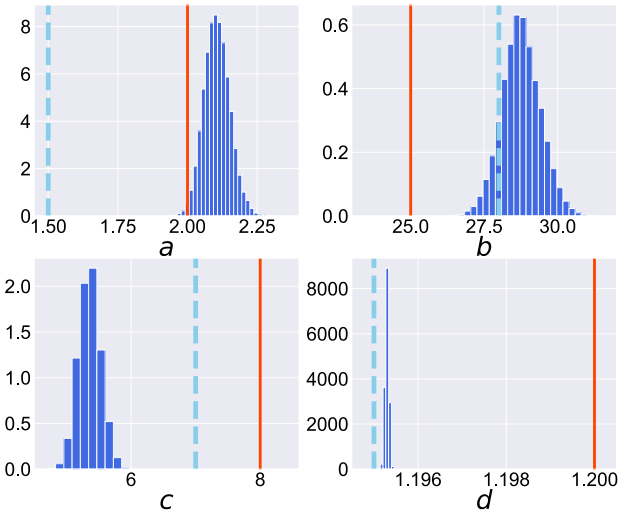

Fig. 5. (Color online) Estimated posterior distributions of parameters obtained by employing PGAS in the benchmark nonlinear state space model. In each graph the vertical axis represents the value of the probability density function and the horizontal axis represents the values of parameters a, b, c, and d. The red solid lines are true values; the light blue dashed lines are initial values.

is $M = 50$. The initial values of latent variables are $z_{1:N}[0] = [0, 0, \ldots, 0]$, and the number of replicas of REPGAS is $R = 90$.

We show the estimated results of the parameters $\boldsymbol{\Theta}$ and latent variables $z_{1:N}$ in Figs. 5, 6, and 7. Figure 5 shows the results of parameters $\boldsymbol{\Theta} = [a, b, c, d]$ estimated with PGAS, and Fig. 6 shows the results estimated with REPGAS. In Figs. 5 and 6, the vertical axis in each graph represents the value of the probability density function and the horizontal axis represents the values of parameters a, b, c, and d. The solid lines are true values, the dashed lines are initial values, and the histograms are estimated posterior distributions of the parameters.

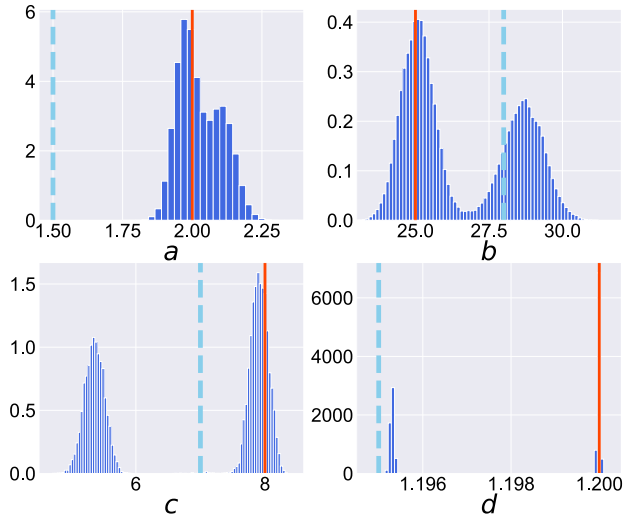

Fig. 6. (Color online) Estimated posterior distributions of parameters obtained by employing REPGAS in the benchmark nonlinear state space model. The red solid lines are true values; the light blue dashed lines are initial values.

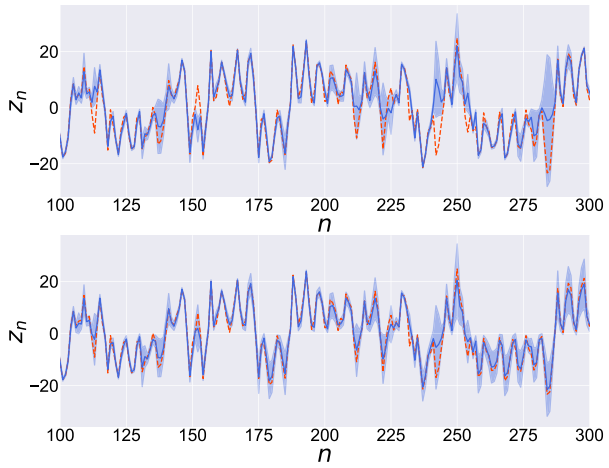

Fig. 7. (Color online) Estimated posterior distributions of latent variables obtained by employing PGAS and REPGAS in the benchmark nonlinear state space model. The upper graph is the result of PGAS and the lower graph is the result of REPGAS. The red dashed lines are true values, the blue solid lines are mean values of estimated distributions, and the blue filled area are the range $\mu \pm \sigma$ (μ is the mean and σ is the standard deviation).

As shown in Fig. 5, the estimated distributions are far from the true values. MCMC methods including PG are guaranteed that the sample from the target distribution can be realized with infinite samples. However, as shown in this result, it may not be possible with a finite number of samples, and in PGAS there is dependence on initial values. In contrast, as shown in Fig. 6, the peak values of the estimated distributions obtained using our REPGAS match the true values. Furthermore, the distributions by REPGAS have multiple peaks in addition to the true ones, and it can be seen from Fig. 6 that PGAS is stuck on the peak that is not true.

Figure 7 shows the estimated results of the latent variables at time steps between $n = 100$ and 300, and the upper graph is the result obtained with PGAS and the lower graph is the result obtained with REPGAS. In both graphs, the vertical axis represents the value of latent variable z_n, and the horizontal axis represents the time step n. The dashed lines are true values, the solid lines are mean values of estimated distributions and the filled areas are the range $\mu \pm \sigma$ (μ is the mean and σ is the standard deviation).

As shown in Fig. 7, in the estimated distributions obtained using PGAS, there are parts where the positive and negative signs of the latent variable z_n are wrong. In consideration of the form of Eq. (18), it is considered that there is a local optimum at the point where the positive and negative signs are reversed. In contrast, since the parameters can be estimated appropriately as shown in Fig. 6, the estimated distributions of latent variables obtained using REPGAS captured the true values.

The above results show that it is possible to overcome the problem of initial value dependence of PGAS by employing REPGAS.

3.2 *Izhikevich neuron model*

Next, to verify the effectiveness of the proposed method, we use the Izhikevich neuron model, which is a computational model of the membrane potential of a neuron:[27,28]

$$\frac{dv}{d\tau} = 0.04v^2 + 5v + 140 - u + I_{\mathrm{ex}},$$
$$\frac{du}{d\tau} = a(bv - u),$$

where v and u are respectively the membrane potential and membrane recovery variable. I_{ex} is the input current from outside the neuron, and a and b are parameters expressing the characteristic of the neuron. If the membrane potential v reaches the threshold v_θ in the Izhikevich neuron model, the membrane potential and membrane recovery variable are reset as follows:

$$v \leftarrow c,$$
$$u \leftarrow u + d,$$

where c and d are also parameters expressing the characteristic of the neuron.

In this paper, assuming that the membrane potential of the Izhikevich neuron model with Gaussian system noise and Gaussian observation noise is observed, we estimate the membrane potential without observation noise $v_{1:N}$, the membrane recovery variables $u_{1:N}$, and parameters $\boldsymbol{\Theta} = \{a, b, c, d\}$ simultaneously from only the observed membrane potential $y_{1:N}$.

In following experiments, we used the true parameters $\boldsymbol{\Theta} = \{a, b, c, d\} = \{0.02, 0.2, -55, 4\}$ and the number of data $N = 200$. Means and variances of the Gaussian system noise superimposed on the membrane potential $v_{1:N}$ and the membrane recovery variable $u_{1:N}$ are $\{\mu_v, \sigma_v^2\} = \{0, 2.25\}$ and $\{\mu_u, \sigma_u^2\} = \{0, 0.25\}$, respectively. The mean and variance of the Gaussian observation noise are $\{\mu_y, \sigma_y\} = \{0, 9\}$.

We show the data used in the following experiments in Fig. 8. In this figure, the vertical axis represents the value of observation y_n and latent variables $\{v_n, u_n\}$, the horizontal axis represents the time step n, the solid line is observation y_n, and the dashed lines are latent variables $\{v_n, u_n\}$.

In REPGAS, the initial values of parameters are $\boldsymbol{\Theta}[0] = [a, b, c, d] = [0.01, 0.8, -65, 8]$ and latent variables are $v_{1:N}[0] = [0, 0, \ldots, 0]$ and $u_{1:N}[0] = [0, 0, \ldots, 0]$. The number of samples is $K = 1.5 \times 10^6$, the number of burn-in

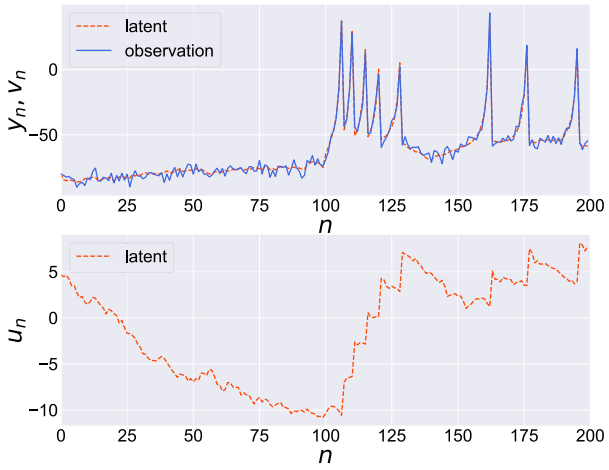

Fig. 8. (Color online) Observations and true latent variables obtained from the Izhikevich neuron model. The vertical axis represents the value of the observation y_n (blue solid line), and two latent variables: the membrane potential v_n (red dashed line), and the membrane recovery variable u_n (red dashed line).

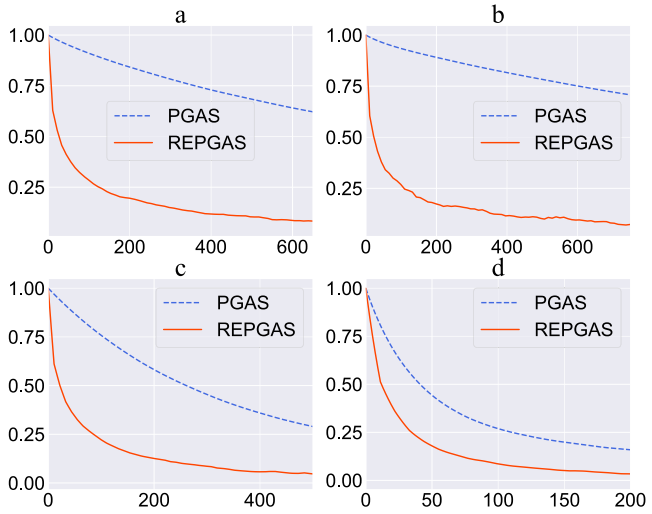

Fig. 9. (Color online) Autocorrelation function of samples of the parameters in the Izhikevich neuron model obtained by employing PGAS and REPGAS. Each line is shown as in Fig. 4.

samples is $K_{\text{burn-in}} = 5 \times 10^5$, the number of particles is $M = 50$, and the number of replicas of REPGAS is $R = 90$.

We show the calculated results of the autocorrelation function with PGAS samples of parameters $\boldsymbol{\Theta}$ and REPGAS samples at $T^1 = 1.0$ in Fig. 9. In all graph, the vertical axis represents the value of the autocorrelation function, and the horizontal axis represents the lag length of the autocorrelation function. The dashed lines are the autocorrelation function results calculated with PGAS samples of parameters a, b, c, and d, and the solid lines are the results calculated with REPGAS samples. It can be confirmed that the sampling efficiency of REPGAS is higher than that of PGAS since the values of the autocorrelation function of REPGAS are lower than those of the autocorrelation function of PGAS when the lag length is small.

Next, we show in Figs. 10 and 11 the estimated results of the posterior distributions of parameters $p(\boldsymbol{\Theta} \mid y_{1:N})$, membrane potentials $p(v_{1:N} \mid y_{1:N})$ and membrane recovery variables $p(u_{1:N} \mid y_{1:N})$ obtained by employing REPGAS.

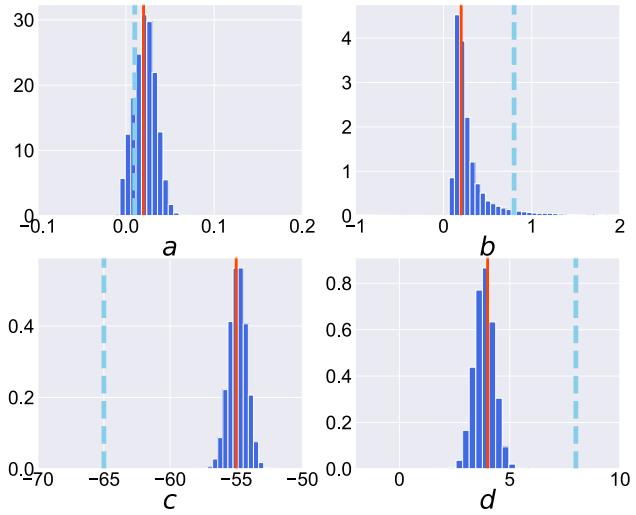

Fig. 10. (Color online) Estimated posterior distributions of parameters in the Izhikevich neuron model obtained by employing REPGAS. The red solid lines are true values; the light blue dashed lines are initial values.

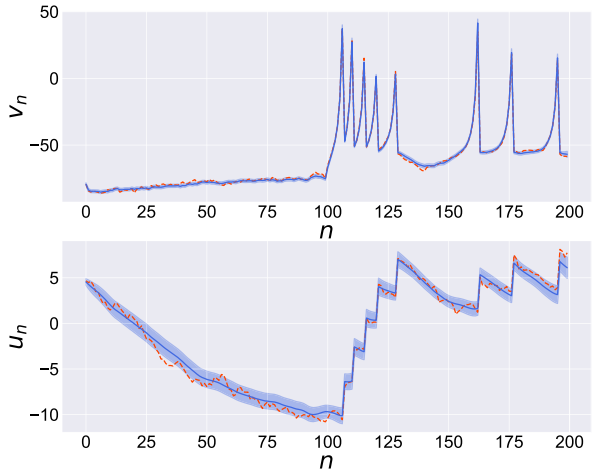

Fig. 11. (Color online) Estimated posterior distributions of latent variables in the Izhikevich neuron model obtained by employing REPGAS. The upper graph is the result of estimated distributions of the membrane potentials $v_{1:N}$ and the lower graph is the result of estimated distributions of the membrane recovery variables $u_{1:N}$. The red dashed lines are true values, the blue lines are mean values of estimated distributions, and the blue filled areas are the range $\mu \pm \sigma$ (μ is the mean and σ is the standard deviation).

Figure 10 shows the estimated results of the posterior distributions of parameters a, b, c, and d. In each graph, the vertical axis represents the value of the probability density function and the horizontal axis represents the values of parameters a, b, c, and d. The solid lines are true values, the dashed lines are initial values, and the histograms are estimated posterior distributions of the parameters. In each of the graphs, it is verified that the true values can be appropriately estimated.

We show the estimated results of latent variables obtained by employing REPGAS in Fig. 11. The upper part of the figure is the result of estimated distributions of the membrane potentials $v_{1:N}$, and the lower part of the figure is the result of estimated distributions of the membrane recovery variables $u_{1:N}$. In both parts of the figure, the vertical axis represents the value of the latent variable and the horizontal axis represents the time step n. The dashed lines are true values,

the solid lines are mean values of estimated distributions, and the filled areas are the range $\mu \pm \sigma$ (μ is the mean and σ is the standard deviation). Both parts of the figure conform the nonlinear changes of the latent variables.

From these results, it was shown that the posterior distribution of not only latent variables $\{v_{1:N}, u_{1:N}\}$ but also parameters $\boldsymbol{\Theta}$ in the Izhikevich neuron model can be estimated more efficiently by employing the proposed method than it can by employing the conventional method. As for the estimation of the posterior distributions of latent variables and parameters by employing PGAS, the distributions converge to the same results as in Figs. 10 and 11.

4. Conclusion

In this paper, we have proposed REPGAS, a method for estimating the joint posterior distribution of latent variables and parameters in a state space model. The problem of initial value dependence in REPGAS is improved compared to PGAS by combining PGAS with the replica exchange method. We have shown in experiments using the benchmark nonlinear state space model that REPGAS can improve the problem of initial value dependence of PGAS and estimate the joint posterior distribution. In addition to improving the problem, REPGAS succeeds in sampling from a multimodal posterior distribution. Furthermore, we have also shown that the autocorrelation time is reduced in the Izhikevich neuron model by employing REPGAS compared to PGAS.

We have proposed in this paper a method for estimating the joint posterior distribution of latent variables and parameters. The proposed concept is a general framework that introduces advantages of the replica exchange method to particle Markov chain Monte Carlo methods. If we are interested in the joint posterior distribution of only parameters marginalized over latent variables, we can employ the particle marginal Metropolis–Hastings (PMMH) algorithm to estimate the distribution.[8] Since the PMMH algorithm also has the problem of initial value dependence, it would be improved by combining it with the replica exchange method, too. Furthermore, it is important to determine the temperature distribution that maximizes the efficiency of the replica exchange method in PGAS and PMMH. We leave this as a future work.

Acknowledgments This work is partially supported by Grants-in-Aid for Scientific Research for Innovative Areas "Initiative for High-Dimensional Data driven Science through Deepening of Sparse Modeling" (JSPS KAKENHI Grant No. JP25120010) and for Scientific Research (JSPS KAKENHI Grant Nos. JP16K00330, JP17H02923, and JP19H04125), and a Fund for the Promotion of Joint International Research (Fostering Joint International Research) (JSPS KAKENHI Grant No. JP15KK0010) from the Ministry of Education, Culture, Sports, Science and Technology of Japan, and CREST (Nos. JPMJCR1755, JPMJCR1861, and JPMJCR1914), Japan Science and Technology Agency, Japan.

*omori@eedept.kobe-u.ac.jp

1) M. L. Andrade Netto, L. Gimeno, and M. J. Mendes, IFAC Proc. Vol. **11**, 2123 (1978).
2) Z. Ghahramani and G. E. Hinton, University of Toronto Technical Report CRG-TR-96-2 (1996).
3) A. Doucet, S. Godsill, and C. Andrieu, Stat. Comput. **10**, 197 (2000).
4) A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer, New York, 2001).
5) R. Meyer and N. Christensen, Phys. Rev. E **65**, 016206 (2001).
6) R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, Int. J. Forecasting **18**, 439 (2002).
7) C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Heidelberg, 2006).
8) C. Andrieu, A. Doucet, and R. Holenstein, J. R. Stat. Soc., Ser. B **72**, 269 (2010).
9) T. Tsunoda, T. Omori, H. Miyakawa, M. Okada, and T. Aonishi, J. Phys. Soc. Jpn. **79**, 124801 (2010).
10) F. Kwasniok, Phys. Rev. E **86**, 036214 (2012).
11) F. Lindsten, T. Schön, and M. I. Jordan, Advances in Neural Information Processing Systems, 2012, p. 25.
12) S. Henriksen, A. Wills, T. B. Schön, and B. Ninness, IFAC Proc. Vol. **45**, 1143 (2012).
13) R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, Advances in Neural Information Processing Systems, 2013, p. 26.
14) F. Lindsten, M. I. Jordan, and T. B. Schön, J. Mach. Learn. Res. **15**, 2145 (2014).
15) T. Omori, T. Kuwatani, A. Okamoto, and K. Hukushima, Phys. Rev. E **94**, 033305 (2016).
16) R. D. Snyder, J. K. Ord, A. B. Koehler, K. R. McLaren, and A. N. Beaumont, Int. J. Forecasting **33**, 502 (2017).
17) T. Omori, T. Sekiguchi, and M. Okada, J. Phys. Soc. Jpn. **86**, 084802 (2017).
18) S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, Advances in Neural Information Processing Systems, 2018, p. 31.
19) P. Gregory, *Bayesian Logical Data Analysis for the Physical Sciences* (Cambridge University Press, Cambridge, U.K., 2010).
20) C. F. J. Wu, Ann. Stat. **11**, 95 (1983).
21) G. J. McLachlan and T. Krishnan, *The EM Algorithm and Its Extensions* (Wiley, New York, 1997).
22) N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, J. Chem. Phys. **21**, 1087 (1953).
23) W. K. Hastings, Biometrika **57**, 97 (1970).
24) S. Geman and D. Geman, IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**, 721 (1984).
25) K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
26) R. Urano and Y. Okamoto, Comput. Phys. Commun. **196**, 380 (2015).
27) E. M. Izhikevich, IEEE Trans. Neural Networks **14**, 1569 (2003).
28) E. M. Izhikevich, IEEE Trans. Neural Networks **15**, 1063 (2004).