Estimating Latent Dynamics for Time-Series Data with Particle Markov Chain Monte Carlo Methods

井上, 広明

(Degree)
博士 (工学)
(Date of Degree)
2022-03-25
(Date of Publication)
2023-03-01
(Resource Type)
doctoral thesis
(Report Number)
甲第8351号
(URL)
https://hdl.handle.net/20.500.14094/D1008351

Doctoral Dissertation

Estimating Latent Dynamics for Time-Series Data with Particle Markov Chain Monte Carlo Methods

（粒子マルコフ連鎖モンテカルロ法を用いた時系列データの潜在ダイナミクス推定に関する研究）

January 2022
Graduate School of Engineering
Kobe University
Hiroaki Inoue
（井上　広明）

Acknowledgements

In this course of writing this dissertation, I have received great deals of help from many people directly and indirectly. My grateful thanks go to all those who supported and encouraged me.

I am extremely grateful to my supervisor, Associate Professor Toshiaki Omori, for his guidance, support and helpful suggestions throughout my bachelor's, master's and doctoral studies.

I would like to express my gratitude to dissertation committee members, Professor Seiichi Ozawa, Professor Tomoya Ono, and Professor Tsutomu Terada, for the suggestions and comments to improve this dissertation.

I would like to thank Professor Seiichi Ozawa, Associate Professor Tomoya Tamei, Project Lecturer Kaien Abe, Assistant Professor Sangwook Kim, Project Assistant Professor Haruka Murakami, Dr. Lihua Wang, and Dr. Jun Kitazono, for willing to give advices and guidances as to improve the quality of my research.

I would like to express my gratitude to Professor Koji Hukushima for helpful advices and suggestions in collaborative researches.

I would like to express my gratitude to my superior in Center for Mathematical and Data Sciences, Professor Masa-Hiko Saito, for his consideration for my doctoral study.

I would like to thank the members in Intelligent Learning Theory Laboratory for their kind help and discussion.

Finally, I would like to express my thanks to my parents and my wife for their

moral support and warm encouragements.

Abstract

In recent years, due to the development of measurement technology, various time series data have been obtained and accumulated in various fields. To understand latent dynamical systems underlying observed time-series data, it is important to estimate latent dynamics from the observed time-series data.

A state space model is a probabilistic model for time-series data that assumes the existence of latent variables which cannot be observed directly. State space models have been used in various fields to forecast observation values and to estimate latent variables. However, model parameters of the state space models are unknown in many cases, and it is important to estimate the parameters from observations.

The particle Markov chain Monte Carlo (PMCMC) methods have been used to estimate latent variables and parameters governing dynamics in state space models from time-series observations. There are two well-known PMCMC methods, the particle-Gibbs (PG) method and the particle marginal Metropolis-Hastings (PMMH) method. Both PMCMC methods consist of Markov chain Monte Carlo (MCMC) methods with the sequential Monte Carlo (SMC) method. The PG method combines the SMC method with Gibbs sampling, and samples latent variables and parameters from the joint posterior distribution of latent variables and parameters alternately. On the other hand, the PMMH method combines the SMC method with the Metropolis-Hastings (MH) algorithm, and samples parameters of state space models from the marginal posterior distribution of parameters. However, both the PG method and the PMMH method are known to have the problems of the initial value dependencies, and cannot estimate the latent dynamical system accurately; the conventional PMCMC methods cannot find the global optimum depending on the initial values in a finite number of samples. Even the extended version of the PG method called the particle-Gibbs with ancestor sampling (PGAS) method, which includes sampling in the backward direction, has the problem of the initial value dependency, and it may not be possible to estimate the latent variables and the parameters precisely depending on the initial values.

In this dissertation, we propose replica exchange particle Markov chain Monte Carlo (REPMCMC) methods for estimating dynamical systems from observable time-series data. We introduce the effect of replica exchange method to probabilistic time-series data analysis in order to realize the local precise search and global search simultaneously for estimating probabilistic state space models. In the proposed method, extended variables corresponding to temperatures are introduced in order to conduct PMCMC methods in parallel at different temperatures. By exchanging samples called replicas between different temperatures, the proposed methods realize the local precise search and global search simultaneously for probabilistic state space models. Firstly, we propose a replica exchange particle Gibbs with ancestor sampling (REPGAS) method for simultaneously estimating distribution of latent variables and model parameters. In the REPGAS method, the exchanges between replicas of pairs of latent variables and model parameters are considered for precisely estimating latent variables and model parameters. Secondly, we propose a replica exchange particle marginal Metropolis-Hastings (REPMMH) method for estimating distribution of model parameters. In the REPMMH method, the exchanges between replicas of model parameters are considered for precisely estimating model parameters.

We verify the effectiveness of proposed REPMCMC methods (the REPGAS method and the REPMMH method) by conducting experiments using a benchmark nonlinear state space model, a nonlinear neuronal model, and a stochastic volatility model. These validation experiments showed that the proposed methods improve the problems of the initial value dependencies as well as the sampling efficiencies of both the conventional PGAS and PMMH methods. These results suggest that the proposed REPGAS method is effective for estimating distribution of latent variables and model parameters of the probabilistic state space model whereas the proposed REPMMH method is effective for estimating distribution of model parameters.

Furthermore, in this dissertation, we apply the REPGAS method to an estimation problem of nonlinear dynamical system with time series of point process. We propose a probabilistic framework for extracting the nonlinear neural dynamics from observable spike-train data of neurons. In this application, we formulate the generative process of the spike-train data as the state space model based on the Izhikevich neuron model, which is known to be able to express a variety of responses of membrane potentials of neurons. We also formulate the process observing the spike-train data according to the latent membrane potential by using the Poisson distribution. We verified that the proposed method precisely estimates the latent variables and the parameters that can represent the important responses of neurons through validation experiments using simulated data. This result shows the proposed framework with the REPGAS method and probabilistic framework from nonlinear dynamics to spike-train data is effective for the inverse problem for nonlinear neural dynamics from spike-train data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, due to the development of measurement technology, various time-series data have been obtained and accumulated in various fields. To understand latent dynamical systems underlying observed time-series data, it is important to estimate latent dynamics from the observed time-series data. A state space model is a probabilistic model for time-series data, that assume the existence of latent variables which cannot be observed directly [1–26]. State space models have been used in various fields to forecast observation values [7, 23, 24] and to estimate latent variables [12, 20, 21, 27]. However, model parameters of the state space models are unknown in many cases. Therefore, it is important to estimate the model parameters of the state space model from observations.

To estimate the parameters of the state space models from observations, a method combining the sequential Monte Carlo (SMC) method [3–5, 8–12, 15–20, 25, 26] with the expectation-maximization (EM) algorithm [8, 22, 28–31] has been proposed [9, 10, 12, 15, 20]. In this approach, the SMC method has been used to estimate the latent variables where as the EM algorithm has been used to estimate the parameters of the state space models. The EM algorithm is based on a maximum likelihood estimation framework and estimates the parameters by sequentially updating the parameters so that the likelihood of the parameters increases. Although it is guaranteed for the EM algorithm that a local optimum can be estimated by iteratively updating the parameters, the global optimum may not be estimated depending on the initial values of the parameters. Furthermore, since the EM algorithm is a point estimation method, it is not possible to identify whether converged values are local or global optima.

To estimate the distribution of parameters of state space models rather than a specific point value of parameters, two kinds of the particle Markov chain Monte Carlo (PMCMC) methods have been proposed: the particle-Gibbs (PG) method and the particle marginal Metropolis-Hastings (PMMH) method [11]. Both PMCMC methods consist of Markov chain Monte Carlo (MCMC) methods with the SMC method, and the distribution of parameters is estimated by collecting samples. The PG method combines the SMC method with Gibbs sampling [8,32–37], and samples latent variables and parameters of state space models alternately from the joint posterior distributions of latent variables and parameters. In the PG method, the SMC method is employed for sampling latent variables. On the other hand, the PMMH method combines the SMC method with the Metropolis-Hastings (MH) algorithm [8,38–42], and samples parameters of state space models from the marginal posterior distributions of parameters. In the PMMH method, the SMC method is employed for calculating the likelihood marginalized over the distribution of latent variables. Both the PG method and the PMMH method have been widely applied (for example, the PG method [43–45], the PMMH method [46–48]).

1.2 Problems

Both the PG method and the PMMH method are known to have the problems of the initial value dependencies, and the samples of the conventional PMCMC methods may be trapped in local optima. In other words, it is difficult to precisely estimate the latent variables and the parameters by employing both the PG method and the PMMH method when the model of interest is very complex. Even the extended version of the PG method called the particle-Gibbs with ancestor sampling (PGAS) method, which includes sampling in the backward direction [16,18,19], also has the problem of the initial value dependency, and it may not be possible to estimate the latent variables and the parameters precisely depending on the initial values. Therefore, it is necessary to develop new methods that can escape the local optima and can precisely find the global optimum for time-series data.

1.3 Purpose of this study

In this dissertation, as new methods to estimate the series of latent variables $\boldsymbol{z}_{1:N}$ and parameters $\boldsymbol{\theta}_{\mathrm{sys}}$ from observable time-series data $\boldsymbol{y}_{1:N}$ in dynamical systems, we propose replica exchange particle Markov chain Monte Carlo (REPMCMC) methods. As shown in Fig 1.1, we construct the framework for accurately estimating latent variables $\boldsymbol{z}_{1:N}$ and parameters $\boldsymbol{\theta}_{\mathrm{sys}}$ from observable time-series data $\boldsymbol{y}_{1:N}$ at time steps $1, 2, \ldots, N$ in dynamical systems. We introduce the effect of replica exchange method [49–51] to probabilistic time-series data analysis in order to realize the local precise search and global search simultaneously for estimating probabilistic state space models. In the proposed method, extended variables corresponding to temperatures are introduced in order to conduct PMCMC methods in parallel at different temperatures. By exchanging samples called as replicas between different temperatures, the proposed methods realize the local precise search and global search simultaneously for probabilistic state space models.

Firstly, we propose a replica exchange particle Gibbs with ancestor sampling (REPGAS) method for simultaneously estimating distributions of latent variables $\boldsymbol{z}_{1:N}$ and model parameters $\boldsymbol{\theta}_{\mathrm{sys}}$ of state space models from observations $\boldsymbol{y}_{1:N}$. In the REPGAS method, the exchanges between replicas of pairs of latent variables $\boldsymbol{z}_{1:N}$ and model parameters $\boldsymbol{\theta}_{\mathrm{sys}}$ are considered for precisely estimating latent variables

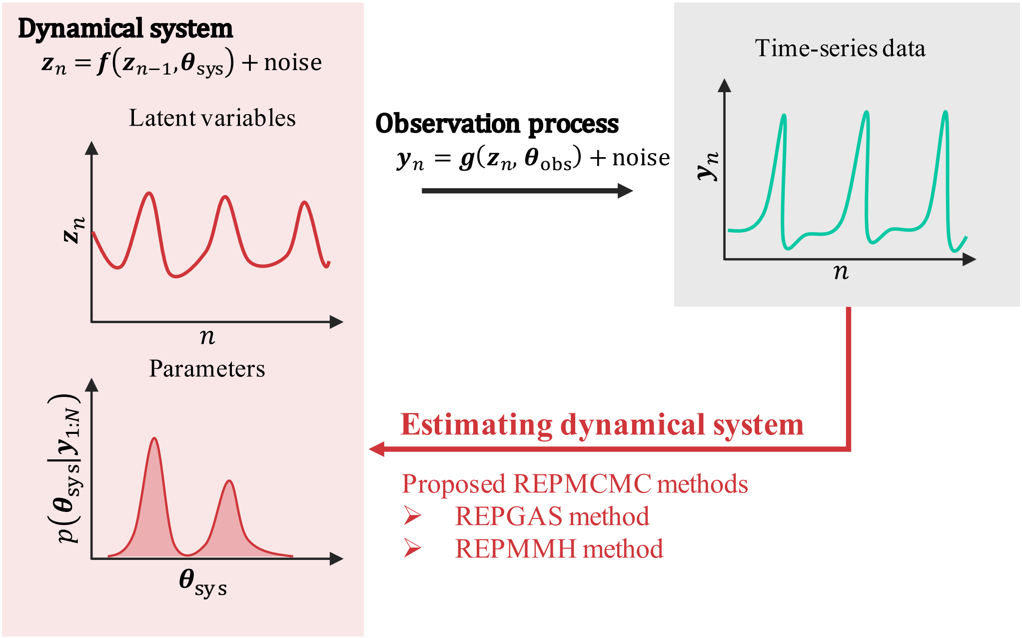

Figure 1.1: The framework for estimating latent variables and parameters from observable time-series data in dynamical systems. In this dissertation, in order to precisely estimate the series of latent variables $\boldsymbol{z}_{1:N}$ and parameters $\boldsymbol{\theta}_{\text{sys}}$ from observed time-series data $\boldsymbol{y}_{1:N}$ in dynamical systems, we propose two kinds of replica exchange particle Markov chain Monte Carlo (REPMCMC) methods: the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method and the replica exchange particle marginal Metropolis-Hastings (REPMMH) method.

$\boldsymbol{z}_{1:N}$ and model parameters $\boldsymbol{\theta}_{\text{sys}}$. Secondly, we propose a replica exchange particle marginal Metropolis-Hastings (REPMMH) method for estimating distribution of model parameters $\boldsymbol{\theta}_{\text{sys}}$. In the REPMMH method, the exchanges between replicas of model parameters $\boldsymbol{\theta}_{\text{sys}}$ are considered for precisely estimating model parameters $\boldsymbol{\theta}_{\text{sys}}$. In the proposed methods, the problems of initial value dependencies in the conventional PMCMC methods are improved due to the global searches, and we can precisely estimate probabilistic state space models.

1.4 Organization of this dissertation

The organization of this dissertation is as follows. In Chapter 2, we describe some Monte Carlo methods for time-series data. We describe the SMC method to estimate latent variables of state space models, and we explain the MCMC methods used to estimate parameters in the PMCMC methods. In Chapter 3, we first describe the PGAS method which is an extended version of the PG method. After that, we propose the REPGAS method to precisely estimate latent variables and parameters of state space models simultaneously by combining the replica exchange method with the PGAS method, and verify the effectiveness of the proposed REPGAS method by using the benchmark nonlinear state space model. In Chapter 4, as an application of the proposed REPGAS method, we propose a method to simultaneously estimate the latent variables and the parameters of the Izhikevich neuron model, which is a mathematical model of neurons, only from limited observations. In Chapter 5, we first describe the PMMH method to estimate marginal posterior distributions of parameters in state space models obtained by marginalized over the distributions of latent variables. After that, we propose the REPMMH method to precisely estimate parameters of state space models by combining the replica exchange method with the PMMH method, and verify the effectiveness of the proposed REPMMH method by using the Izhikevich neuron model and a stochastic volatility model. Finally, the conclusion is given in Chapter 6.

Chapter 2

Monte Carlo methods for time-series data

In this chapter, we explain the Monte Carlo methods used to estimate latent variables and parameters of state space models in this dissertation. First, we explain probabilistic models for state space models and the sequential Bayesian filtering for estimating latent variables of state space models. After that, we describe the algorithm of the sequential Monte Carlo (SMC) method for obtaining the posterior distribution of latent variables approximately. Next, we describe the Markov chain Monte Carlo (MCMC) methods used to estimate parameters of state space models in this dissertation. We explain the conditions to satisfy in the MCMC methods, and introduce the Gibbs sampling and the Metropolis-Hastings (MH) algorithm, which are widely used in the MCMC methods.

2.1 Sequential Monte Carlo method

The state space model is a general model to represent a generative process of time-series data, and assumes that there are time-varying latent variables in the background of obtaining observations [1–26]. In various fields, the state space model has used to forecast future observations [7, 23, 24] and estimate latent variables [12, 20, 21, 27].

The state space model consists of two models called a system model and an observation model. The system model represents the time evolution of latent variables, and the observation model expresses the process of obtaining the observations from the latent variables. In this section, we first describe the probabilistic models of the state space model. Next, we explain three kinds of posterior distributions of the latent variables and the sequential Bayesian filtering for estimating these posterior distributions. After that, we describe the SMC method, which is a typical method to estimate the posterior distributions of the latent variables from the observations approximately.

2.1.1 State space model

Let us denote the latent variables by $\boldsymbol{z}_n$ and the observations by $\boldsymbol{y}_n$ at time step $n \in \{1, 2, \ldots, N\}$. Figure 2.1 shows the state space model as a graphical model. Here, in Fig. 2.1, the arrow from $\boldsymbol{z}_{n-1}$ to $\boldsymbol{z}_n$ represents a system model, and the arrow from $\boldsymbol{z}_n$ to $\boldsymbol{y}_n$ represents an observation model. The system model and the observation model at time step n are expressed as follows:

$$\boldsymbol{z}_n \sim p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\mathrm{sys}}\right), \tag{2.1}$$

$$\boldsymbol{y}_n \sim p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right), \tag{2.2}$$

where $\boldsymbol{\theta}_{\mathrm{sys}}$ and $\boldsymbol{\theta}_{\mathrm{obs}}$ are the parameters in the system model and the observation model, respectively.

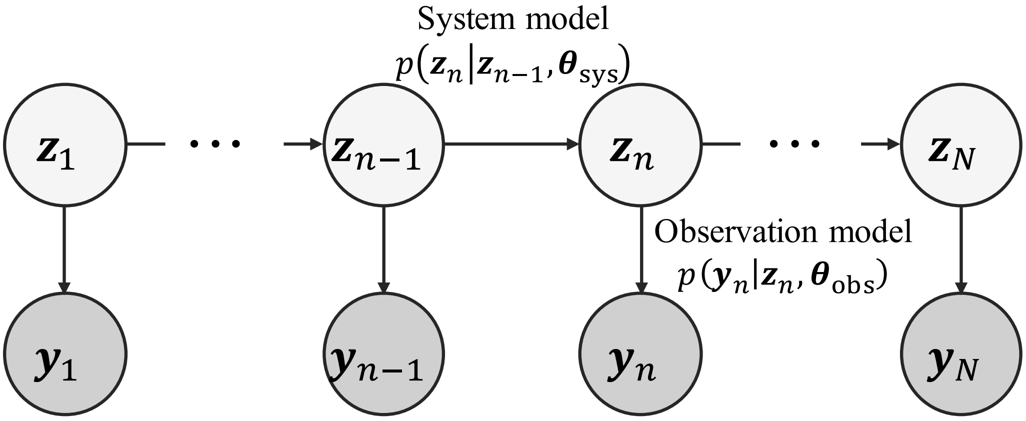

Figure 2.1: Probabilistic graphical model of a state space model. $\boldsymbol{z}_{1:N} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ and $\boldsymbol{y}_{1:N} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$ respectively represent latent variables and observations for time step $n = 1, 2, \ldots, N$. The arrow to the latent variables $\boldsymbol{z}_n$ at the time step n from the latent variables $\boldsymbol{z}_{n-1}$ at the previous time step $n-1$ represents a system model $p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\text{sys}}\right)$, and the arrow to the observations $\boldsymbol{y}_n$ at the time step n from the latent variables $\boldsymbol{z}_n$ at the time step n represents an observation model $p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\text{obs}}\right)$. $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\text{sys}}, \boldsymbol{\theta}_{\text{obs}}\}$ are parameters.

2.1.2 Sequential Bayesian filtering for estimating latent variables

To estimate latent dynamics, we consider posterior distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:l}, \boldsymbol{\Theta}\right)$ of latent variables $\boldsymbol{z}_n$ at time step n given the series of observations $\boldsymbol{y}_{1:l}$ at the time steps $1, 2, \ldots, l$ and parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\text{sys}}, \boldsymbol{\theta}_{\text{obs}}\}$. Depending on the relationship between time steps n and l, we consider three kinds of posterior distributions: the predictive distribution, the filtering distribution, or the smoothing distribution. These posterior distributions $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:l}, \boldsymbol{\Theta}\right)$ of latent variables $\boldsymbol{z}_n$ can be calculated sequentially with the sequential Bayesian filtering as follows.

The distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:l}, \boldsymbol{\Theta}\right)$ of latent variables at time step n given observations up to previous time step l $(l < n)$ is called the predictive distribution. In particular, when $l = n-1$, it is called the one-step-ahead predictive distribution

$p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ and expressed with the system model $p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\mathrm{sys}}\right)$ as follows:

$$p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right) = \int p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\mathrm{sys}}\right) p\left(\boldsymbol{z}_{n-1} \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right) d\boldsymbol{z}_{n-1}, \tag{2.3}$$

where the distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ of latent variables at time step n given observations up to the same time step n $(l = n)$ is called the filtering distribution, in order to obtain the one-step-ahead predictive distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ at the time step n, the filtering distribution $p\left(\boldsymbol{z}_{n-1} \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ at the previous time step $n-1$ is needed.

The filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ at the time step n is represented with the observation model $p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right)$ as follows:

$$p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right) = \frac{p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right) p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)}{\int p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right) p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right) d\boldsymbol{z}_n}, \tag{2.4}$$

where the one-step-ahead predictive distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ at the time step n is needed in order to obtain the filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ at the time step n. In other words, by iterating to calculate the one-step-ahead predictive distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ and the filtering distribution $p\left(\boldsymbol{z}_{n-1} \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ at each time step $n \in \{1, 2, \ldots, N\}$ alternately, the filtering distributions for all time steps can be obtained.

If the time step l is more than n, the distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:l}, \boldsymbol{\Theta}\right)$ is called the smoothing distribution. When $l = N$, the smoothing distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right)$ $(n < N)$ at the time step n is obtained as follows:

$$p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right) = p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right) \int \frac{p\left(\boldsymbol{z}_{n+1} \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right) p\left(\boldsymbol{z}_{n+1} \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{sys}}\right)}{p\left(\boldsymbol{z}_{n+1} \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)} d\boldsymbol{z}_{n+1}, \tag{2.5}$$

where the one-step-ahead predictive distribution $p\left(\boldsymbol{z}_{n+1} \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ at the next time step $n+1$, the filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ at the time step n, and the smoothing distribution $p\left(\boldsymbol{z}_{n+1} \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right)$ at the next time step $n+1$ are needed in

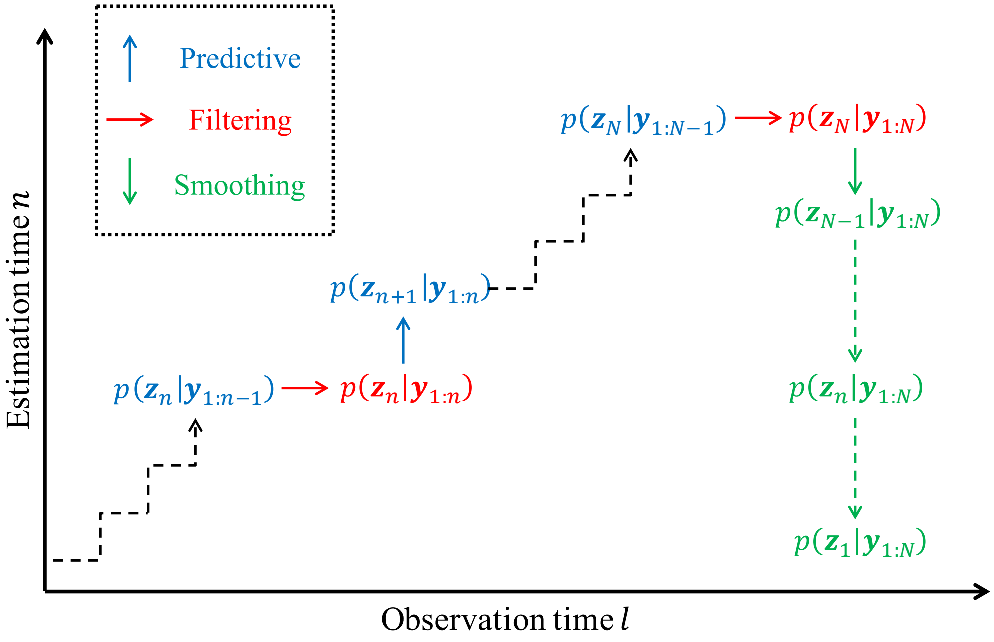

Figure 2.2: The flow of the sequential Bayesian filtering to obtain the predictive, filtering, and smoothing distributions. The predictive distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$ and filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ need to be obtained alternately, and the smoothing distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right)$ needs to be obtained sequentially in the backward direction after obtaining the filtering distribution at the time step N.

order to obtain the smoothing distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right)$ at the time step n. In brief, in order to obtain the smoothing distributions at all time steps, it is necessary to calculate the smoothing distributions retroactively from the time step N to 1 after alternately calculating the one-step-ahead predictive distribution and the filtering distribution from the time step 1 to N.

We show the flow of the sequential Bayesian filtering to obtain the predictive, filtering, and smoothing distributions in Fig. 2.2. The predictive and filtering distributions need to be obtained alternately, and the smoothing distributions need to be obtained sequentially in the backward direction after obtaining the filtering distribution at the time step N.

Algorithm 1 Sequential Monte Carlo (SMC) method

```
generate initial particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \ldots, \hat{\boldsymbol{z}}_1^{(M)}\right\}$ from initial distribution $p\left(\boldsymbol{z}_1\right)$ by Eq. (2.6)
calculate likelihoods as the weights of the particles $\left\{w_1^{(1)}, w_1^{(2)}, \ldots, w_1^{(M)}\right\}$ by Eq. (2.7)
normalize the weights of particles $\left\{W_1^{(1)}, W_1^{(2)}, \ldots, W_1^{(M)}\right\}$ by Eq. (2.8)
resample particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \ldots, \hat{\boldsymbol{z}}_1^{(M)}\right\}$ according to the normalized weights $\left\{W_1^{(1)}, W_1^{(2)}, \ldots, W_1^{(M)}\right\}$
for $n = 2, \ldots, N$ do
    update particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \ldots, \hat{\boldsymbol{z}}_n^{(M)}\right\}$ for prediction by Eq. (2.9)
    calculate likelihoods as the weights of the particles $\left\{w_n^{(1)}, w_n^{(2)}, \ldots, w_n^{(M)}\right\}$ by Eq. (2.10)
    normalize the weights of the particles $\left\{W_n^{(1)}, W_n^{(2)}, \ldots, W_n^{(M)}\right\}$ by Eq. (2.11)
    resample particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \ldots, \hat{\boldsymbol{z}}_n^{(M)}\right\}$ according to the normalized weights $\left\{W_n^{(1)}, W_n^{(2)}, \ldots, W_n^{(M)}\right\}$
end for
```

2.1.3 Sequential Monte Carlo method

The SMC method is also called the particle filter, and it is a method for obtaining the filtering distribution approximately using particles with realized values of latent variables [3–5, 8–12, 15–20, 25, 26]. We show the algorithm of the SMC method in Algorithm 1 and the flow of the SMC method in Fig. 2.3. In the SMC method, the filtering distribution can be approximated by iterating the three processes of generating or updating the particles, calculating the weights of the particles, and resampling the particles at each time step.

When the time step $n = 1$, particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \ldots, \hat{\boldsymbol{z}}_1^{(M)}\right\}$ are generated using the initial distribution of latent variables $p\left(\boldsymbol{z}_1\right)$. Here, M is the number of particles, and i-th particle is generated as follows:

$$\hat{\boldsymbol{z}}_1^{(i)} \sim p\left(\boldsymbol{z}_1\right). \tag{2.6}$$

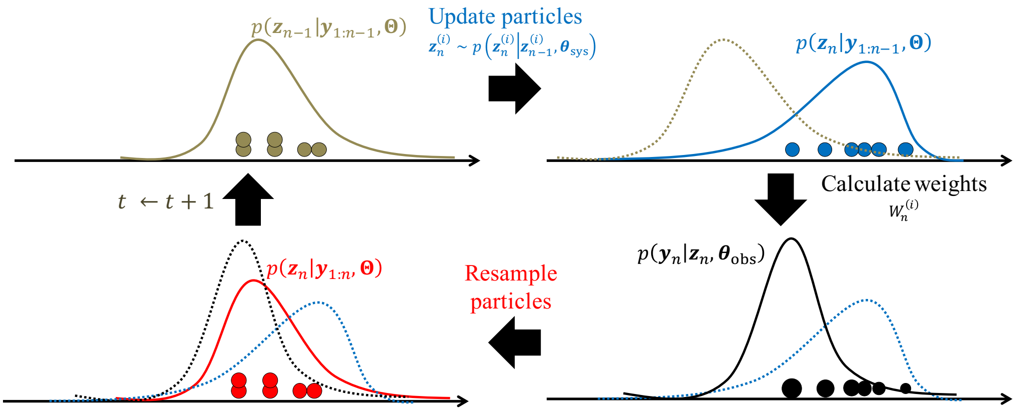

Figure 2.3: The flow of the sequential Monte Carlo (SMC) method. In the SMC method, the filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ can be approximated by iterating the three processes of updating the particles, calculating the weights of the particles, and resampling the particles at each time step.

Next, the weights $w_1^{(i)}$ of each particle is calculated using the observation model $p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\text{obs}}\right)$ as follows:

$$w_1^{(i)} = p\left(\boldsymbol{y}_1 \,\middle|\, \hat{\boldsymbol{z}}_1^{(i)}, \boldsymbol{\theta}_{\text{obs}}\right). \tag{2.7}$$

Moreover, we normalize the weights of the particles as follows:

$$W_1^{(i)} = \frac{w_1^{(i)}}{\sum_{j=1}^{M} w_1^{(j)}}. \tag{2.8}$$

Finally, we resample the particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \ldots, \hat{\boldsymbol{z}}_1^{(M)}\right\}$ according to the normalized weights $\left\{W_1^{(1)}, W_1^{(2)}, \ldots, W_1^{(M)}\right\}$.

When $n = 2, 3, \ldots, N$, we update the particles using the system model $p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\text{sys}}\right)$ as follows:

$$\hat{\boldsymbol{z}}_n^{(i)} \sim p\left(\hat{\boldsymbol{z}}_n^{(i)} \,\middle|\, \hat{\boldsymbol{z}}_{n-1}^{(i)}, \boldsymbol{\theta}_{\text{sys}}\right). \tag{2.9}$$

Moreover, we calculate the weights of particles using the observation model $p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\text{obs}}\right)$

and normalize the weights of particles as follows:

$$w_n^{(i)} = p\left(\boldsymbol{y}_n \;\middle|\; \hat{\boldsymbol{z}}_n^{(i)}, \boldsymbol{\theta}_{\mathrm{obs}}\right), \tag{2.10}$$

$$W_n^{(i)} = \frac{w_n^{(i)}}{\sum_{j=1}^{M} w_n^{(j)}}. \tag{2.11}$$

Finally, we resample the particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \ldots, \hat{\boldsymbol{z}}_n^{(M)}\right\}$ according to the normalized weights $\left\{W_n^{(1)}, W_n^{(2)}, \ldots, W_n^{(M)}\right\}$.

In the SMC method, the particles updated with the system model $p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\mathrm{sys}}\right)$ in Eq. (2.9) are samples from the one-step-ahead predictive distributions $p\left(\boldsymbol{z}_n \;\middle|\; \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right)$, and the particles resampled with the normalized weights $\left\{W_n^{(1)}, W_n^{(2)}, \ldots, W_n^{(M)}\right\}$ calculated by the observation model $p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right)$ in Eqs. (2.10) and (2.11) are samples from the filtering distribution $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$. Moreover, by iterating the above processes at all time steps, the filtering distribution at each time step is obtained approximately as follows:

$$p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right) \simeq \frac{1}{M} \sum_{i=1}^{M} \delta\left(\boldsymbol{z}_n - \hat{\boldsymbol{z}}_n^{(i)}\right), \tag{2.12}$$

where $\delta\left(\boldsymbol{z}_n\right)$ is the Dirac delta distribution.

2.2 Markov chain Monte Carlo methods

The MCMC methods are used in the field of statistical physics to obtain samples from arbitrary probability distributions [8, 32–42, 49–51]. In recent years, the MCMC methods have been utilized for optimization problems in machine learning [33, 34, 37, 40–42]. In the MCMC method, samples are obtained according to the transition probabilities satisfying certain conditions, which allows us to indirectly obtain samples from the target distribution for which we want to obtain samples. It is known that we can reproduce the target distribution by collecting a sufficient number of samples obtained by the MCMC method. In this dissertation, we apply the MCMC methods to estimate the parameters of state-space models. In this section, we describe the conditions to satisfy in the MCMC methods, and then introduce the algorithms that satisfy these conditions, the Metropolis-Hastings (MH) algorithm and Gibbs sampling.

2.2.1 Conditions to be satisfied in the MCMC method

In the MCMC methods [8, 32–42, 49–51], to generate the sample set $\{\boldsymbol{X}\}$ from the target distribution $\pi(\boldsymbol{X})$, we constitute a Markov chain of the random variables $\boldsymbol{X}$. The Markov chain is defined by determining the initial state $\boldsymbol{X}[0]$ of the random variables $\boldsymbol{X}$ and the transition probability $W(\boldsymbol{X} \to \boldsymbol{X}')$ between states $\boldsymbol{X}$ and $\boldsymbol{X}'$. In the MCMC methods, it is guaranteed that the sample set $\{\boldsymbol{X}\}$ from the constituted Markov chain converges to the target distribution $\pi(\boldsymbol{X})$ when the transition probability $W(\boldsymbol{X} \to \boldsymbol{X}')$ satisfies following (a) detailed balance condition and (b) ergodic condition [8].

(a) Detailed balance condition

For any two states $\boldsymbol{X}$ and $\boldsymbol{X}'$, the transition probability $W(\boldsymbol{X} \to \boldsymbol{X}')$ satisfy the following equality:

$$\pi(\boldsymbol{X})\, W\left(\boldsymbol{X} \to \boldsymbol{X}'\right) = \pi\left(\boldsymbol{X}'\right) W\left(\boldsymbol{X}' \to \boldsymbol{X}\right). \tag{2.13}$$

(b) Ergodic condition

The transition probability between any two states $\boldsymbol{X}$ and $\boldsymbol{X}'$ is expressed as the product of a finite number of non-zero transition probabilities.

In the MCMC method, it is necessary to design the transition probabilities $W(\boldsymbol{X} \to \boldsymbol{X}')$ to satisfy the detailed balance condition and the ergodic condition, and there are well-known algorithms to satisfy the detailed equilibrium condition, such as the MH algorithm and Gibbs sampling.

Algorithm 2 Metropolis-Hastings (MH) algorithm

```
initialize $\boldsymbol{X}[0]$
$k \leftarrow 1$
repeat
    sample a new sample candidate $\boldsymbol{X}' \sim q(\boldsymbol{X}' | \boldsymbol{X}[k-1])$
    calculate the acceptance probability $a = A(\boldsymbol{X}, \boldsymbol{X}')$
    $\boldsymbol{X}[k] \leftarrow \boldsymbol{X}'$ with the acceptance probability $a$, otherwise $\boldsymbol{X}[k] \leftarrow \boldsymbol{X}[k-1]$
    $k \leftarrow k + 1$
until $k \leq K \{K :$ number of iteration$\}$
```

2.2.2 Metropolis-Hastings algorithm

In the MH algorithm [8, 38–42], we express the trainsition probability $W(\boldsymbol{X} \to \boldsymbol{X}')$ as the product of two probabilities, the proposal probability $q(\boldsymbol{X}' \mid \boldsymbol{X})$ and the acceptance probability $A(\boldsymbol{X}, \boldsymbol{X}')$, as follows:

$$W\left(\boldsymbol{X} \to \boldsymbol{X}'\right) = q\left(\boldsymbol{X}' \mid \boldsymbol{X}\right) A\left(\boldsymbol{X}, \boldsymbol{X}'\right). \tag{2.14}$$

Here, $q(\boldsymbol{X}' \mid \boldsymbol{X})$ represents the probability that the random variables $\boldsymbol{X}'$ are proposed from the random variables $\boldsymbol{X}$. The acceptance probability $A(\boldsymbol{X}, \boldsymbol{X}')$ is represented with the proposal probability $q(\boldsymbol{X}' \mid \boldsymbol{X})$ as follows:

$$A\left(\boldsymbol{X}, \boldsymbol{X}'\right) = \min\left(1, \frac{q(\boldsymbol{X} \mid \boldsymbol{X}')\,\pi(\boldsymbol{X}')}{q(\boldsymbol{X}' \mid \boldsymbol{X})\,\pi(\boldsymbol{X})}\right). \tag{2.15}$$

Here, we show the flow of the MH algorithm in Algorithm 2. In the MH algorithm, the transition probability $W(\boldsymbol{X} \to \boldsymbol{X}')$ are represented by proposing a new sample candidate $\boldsymbol{X}'$ from the current state $\boldsymbol{X}$ according to the proposal probability $q(\boldsymbol{X}' \mid \boldsymbol{X})$ and deciding whether to accept or reject the sample candidate $\boldsymbol{X}'$ with the acceptance probability $A(\boldsymbol{X}, \boldsymbol{X}')$. As shown in Fig. 2.4, in the MH algorithm, the sample from the target distribution $\pi(X)$ is obtained indirectly by proposing sample candidates and deciding whether to accept or reject them repeatedly.

In the MH algorithm, by using arbitrary proposal probability $q(\boldsymbol{X}' \mid \boldsymbol{X})$, the

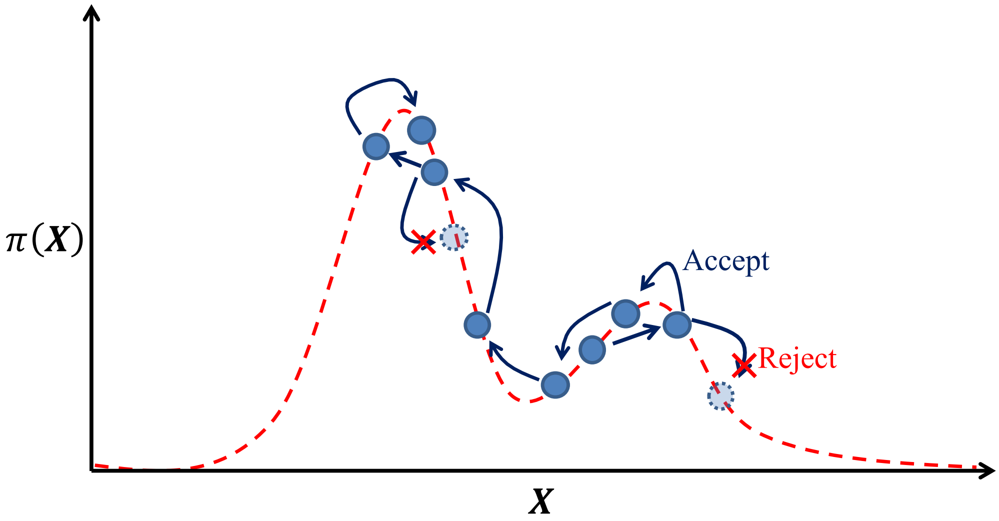

Figure 2.4: The diagram of Metropolis-Hastings (MH) algorithm. In the MH algorithm, the sample from the target distribution $\pi(X)$ is obtained indirectly by proposing sample candidates and deciding whether to accept or reject them repeatedly.

detailed balance condition [Eq. (2.13)] is satisfied as follows:

$$\begin{aligned}
\pi(\boldsymbol{X}) W(\boldsymbol{X} \to \boldsymbol{X}') &= \pi(\boldsymbol{X})\, q(\boldsymbol{X}'|\boldsymbol{X})\, A(\boldsymbol{X}, \boldsymbol{X}') \\
&= \min\left(q(\boldsymbol{X}'|\boldsymbol{X})\, \pi(\boldsymbol{X}), q(\boldsymbol{X}|\boldsymbol{X}')\, \pi(\boldsymbol{X}')\right) \\
&= \pi(\boldsymbol{X}')\, q(\boldsymbol{X}|\boldsymbol{X}') \min\left(\frac{q(\boldsymbol{X}'|\boldsymbol{X})\, \pi(\boldsymbol{X})}{q(\boldsymbol{X}|\boldsymbol{X}')\, \pi(\boldsymbol{X}')}, 1\right) \\
&= \pi(\boldsymbol{X}')\, q(\boldsymbol{X}|\boldsymbol{X}')\, A(\boldsymbol{X}', \boldsymbol{X}) \\
&= \pi(\boldsymbol{X}')\, W(\boldsymbol{X}' \to \boldsymbol{X}) .
\end{aligned}$$

In other words, the sample set $\{\boldsymbol{X}\}$ obtained by the MH algorithm converges to the target distribution $\pi(\boldsymbol{X})$ by using the proposal probability $q(\boldsymbol{X}' \mid \boldsymbol{X})$ satisfying the ergodic condition.

2.2.3 Gibbs sampling

Next, we describe the Gibbs sampling [8, 32–37] which is one of the MCMC methods in this subsection. Here we consider L dimensional random variables $\boldsymbol{X} = [X_1, X_2, \ldots, X_l, \ldots, X_L]$. In the Gibbs sampling, we use the following conditional probability $p\left(X_l \mid \boldsymbol{X}_{\setminus l}\right)$ of the l-th variable X_l of the random variables $\boldsymbol{X}$ conditioned by the other variables $\boldsymbol{X}_{\setminus l} = [X_1, X_2, \ldots, X_{l-1}, X_{l+1}, \ldots, X_L]$ in $\boldsymbol{X}$ as the proposal probability $q\left(\boldsymbol{X}' \mid \boldsymbol{X}\right)$ in the MH algorithm:

$$p\left(X_l \mid \boldsymbol{X}_{\setminus l}\right) = \frac{\pi\left(\boldsymbol{X}\right)}{p\left(\boldsymbol{X}_{\setminus l}\right)} = \frac{\pi\left(X_l, \boldsymbol{X}_{\setminus l}\right)}{p\left(\boldsymbol{X}_{\setminus l}\right)} \; (l = 1, 2, \ldots, L), \tag{2.16}$$

where $p\left(\boldsymbol{X}_{\setminus l}\right)$ is the joint probability of the variables $\boldsymbol{X}_{\setminus l}$.

As shown in Fig. 2.5, in the Gibbs sampling, we propose a sample of each variable X_l in the random variables $\boldsymbol{X}$ according to the conditional probability $p\left(X_l \mid \boldsymbol{X}_{\setminus l}\right)$ in turn while keeping the other variables $\boldsymbol{X}_{\setminus l}$ fixed. Then, the acceptance probability $A\left(\boldsymbol{X}, \boldsymbol{X}'\right)$ of the MH algorithm are calculated as follows:

$$\begin{aligned}
A\left(\boldsymbol{X}, \boldsymbol{X}'\right) &= \min\left(1, \frac{q\left(\boldsymbol{X} \mid \boldsymbol{X}'\right)\pi\left(\boldsymbol{X}'\right)}{q\left(\boldsymbol{X}' \mid \boldsymbol{X}\right)\pi\left(\boldsymbol{X}\right)}\right) \\
&= \min\left(1, \frac{\pi\left(X_l, \boldsymbol{X}_{\setminus l}\right)}{p\left(\boldsymbol{X}_{\setminus l}\right)} \frac{p\left(\boldsymbol{X}_{\setminus l}\right)}{\pi\left(X_l', \boldsymbol{X}_{\setminus l}\right)} \frac{\pi\left(\boldsymbol{X}'\right)}{\pi\left(\boldsymbol{X}\right)}\right) \\
&= \min\left(1, \frac{\pi\left(X_l, \boldsymbol{X}_{\setminus l}\right)}{\pi\left(X_l', \boldsymbol{X}_{\setminus l}\right)} \frac{\pi\left(X_l', \boldsymbol{X}_{\setminus l}\right)}{\pi\left(X_l, \boldsymbol{X}_{\setminus l}\right)}\right) \\
&= \min\left(1, 1\right) \\
&= 1,
\end{aligned}$$

where $\boldsymbol{X}' = \left\{X_l', \boldsymbol{X}_{\setminus l}\right\}$. In other words, if a sample candidate $\boldsymbol{X}' = \left\{X_l', \boldsymbol{X}_{\setminus l}\right\}$ is proposed from the conditional distribution $p\left(X_l \mid \boldsymbol{X}_{\setminus l}\right)$ by using Gibbs sampling, the sample candidate $\boldsymbol{X}' = \left\{X_l', \boldsymbol{X}_{\setminus l}\right\}$ is always accepted in the MH algorithm. Furthermore, since the Gibbs sampling satisfies the criteria of the MH algorithm, the Gibbs sampling also satisfies the detailed balance condition.

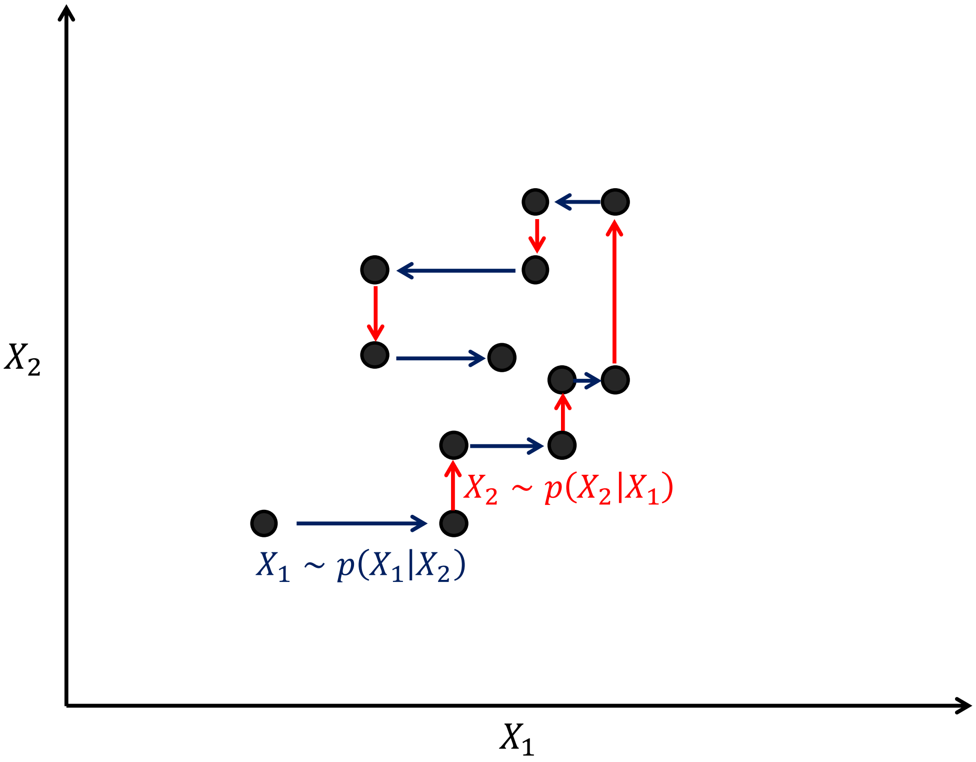

Figure 2.5: The diagram of the Gibbs sampling. In the Gibbs sampling, we propose a sample of each variable X_l in the random variables $\boldsymbol{X}$ according to the conditional probability $p\left(X_l \mid \boldsymbol{X}_{\setminus l}\right)$ in turn while keeping the other variables $\boldsymbol{X}_{\setminus l}$ fixed.

2.3 Summary

In this chapter, we explained the SMC method and the MCMC methods used to estimate latent variables $\boldsymbol{z}_{1:N}$ and parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\mathrm{sys}}, \boldsymbol{\theta}_{\mathrm{obs}}\}$ of state space models. The SMC method approximately estimate the filtering distributions of latent variables $p\left(\boldsymbol{z}_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{\Theta}\right)$ at each time step by using particles with realized values of latent variables $\boldsymbol{z}_{1:N}$. The MCMC methods can obtain samples from the target distribution $\pi\left(\boldsymbol{X}\right)$ by using the transition probability $W\left(\boldsymbol{X} \rightarrow \boldsymbol{X}'\right)$ satisfying the two conditions: the detailed balance condition and the ergodic condition.

In this dissertation, we propose new methods to estimate latent variables and parameters of state space models based on the particle Markov chain Monte Carlo

(PMCMC) methods. The PMCMC methods consist of the SMC method and the MCMC methods, and can estimate latent variables and parameters of state space models. In the PMCMC methods, the SMC method is used to estimate latent variables, and the MCMC methods are used to estimate parameters mainly. In Chapters 3, 4, and 5, we propose new PMCMC methods and its application, which more precisely estimate latent variables and parameters of state space models.

Chapter 3

Replica exchange particle-Gibbs with ancestor sampling method

3.1 Introduction

The particle-Gibbs (PG) method is one of the particle Markov chain Monte Carlo (PMCMC) method for sampling from the joint posterior distribution of latent variables and parameters in a state space model [11]. The PG method combine the sequential Monte Carlo (SMC) method [3–5, 8–12, 15–20, 25, 26] with the Gibbs sampling [8, 32–37], and the latent variables and the parameters are sampled alternately in the PG method. Note that, the SMC method is used to sample the latent variables. However, the SMC method has the problem of degeneracy, and the PG method is affected by the problem of degeneracy in the SMC method. As a result, the sampling efficiency of the PG method becomes low when the number of particles in the SMC method is small. Moreover, the PG method guarantees that it is possible to sample from the joint posterior distribution of the latent variables and the parameters in an infinite number of trials, but there also is the problem that it depends on the initial values in a finite number of trials.

The particle-Gibbs with ancestor sampling (PGAS) method has been proposed to overcome the problem of degeneracy in the PG method [16, 18, 19]. The PGAS method is a method to sample the latent variables not only in the forward direction but also in the retroactive direction in the PG method. By considering the retroactive direction, the latent variables of the early time steps, which tend to be degenerate in the SMC method, can be efficiently sampled. By using the PGAS method, the problem of degeneracy and the sampling efficiency in the PG method can be improved. However, similar to the PG method, the PGAS method also has the problem of initial value dependence in a finite number of trials.

In the Markov chain Monte Carlo (MCMC) methods, the replica exchange method has been proposed in order to improve the problem of initial value dependence [49–51]. The replica exchange method introduces an extension variables called temperatures and samples by the MCMC methods at multiple temperatures in parallel. The temperature is extension variable that adjusts the complexity of the target distribution. When the temperature becomes high, distribution approaches to a uniform distribution. Therefore, at high temperature, it is possible to search a wide range without depending on the initial values. In the replica exchange method, the problem of initial value dependence in the MCMC method can be improved by exchanging samples between different temperatures and allowing samples to pass through high temperatures.

In this chapter, we propose the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method combining the PGAS method with the replica exchange method in order to improve the problem of initial value dependence in the PGAS method. We first describe the PGAS method for estimating the joint posterior distribution of the latent variables and the parameters in the state space model. After that, we explain the proposed REPGAS method, and we conduct the experiments using the nonlinear benchmark model in order to verify the effectiveness of the proposed REPGAS method.

Algorithm 3 Particle-Gibbs with ancestor sampling (PGAS) method

```
initialize the samples of the latent variables $\boldsymbol{z}_{1:N}[0]$
initialize the samples of the parameters $\boldsymbol{\Theta}[0]$
for $k = 1, \ldots, K$ do
    sample the latent variables $\boldsymbol{z}_{1:N}[k]$ by employing the CSMCAS method in Algorithm 4
    sample the parameters $\boldsymbol{\Theta}[k] \sim p(\boldsymbol{\Theta} \mid \boldsymbol{z}_{1:N}[k], \boldsymbol{y}_{1:N})$ by employing a MCMC method (e.g., the MH algorithm [8,38–42])
end for
```

3.2 Conventional method: particle-Gibbs with ancestor sampling method

We explain the PGAS method for estimating the joint posterior distribution $p(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N})$ of the latent variables $\boldsymbol{z}_{1:N}$ and the parameters $\boldsymbol{\Theta}$ in a state space model. We show the algorithm of the PGAS method in Algorithm 3. In the PGAS method, the latent variables $\boldsymbol{z}_{1:N}$ and the parameters $\boldsymbol{\Theta}$ are initialized as $\boldsymbol{z}_{1:N}[0]$ and $\boldsymbol{\Theta}[0] = \{\boldsymbol{\theta}_{\mathrm{sys}}[0], \boldsymbol{\theta}_{\mathrm{obs}}[0]\}$, and the samples of latent variables are obtained by the conditional sequential Monte Carlo with ancestor sampling (CSMCAS) method and the samples of parameters are obtained by an MCMC method (e.g., the MH algorithm [8,38–42]) alternately.

In the PGAS method, the k-th sample of latent variables $\boldsymbol{z}_{1:N}[k]$ is obtained by employing the CSMCAS method. At k-th step, the CSMCAS method uses the observations $\boldsymbol{y}_{1:N}$ and the previous sample of parameters $\boldsymbol{\Theta}[k-1]$ as well as the previous sample of latent variables $\boldsymbol{z}_{1:N}[k-1]$ as the conditioned sample of latent variables. The CSMCAS method use the previous sample $\boldsymbol{z}_{1:N}[k-1]$ as a conditional particle in the SMC method. Moreover, when the particles are resampled in the SMC method, the indices of the particles that are the sources of resampling are expressed by the ancestor indices $\left\{A_n^{(1)}, A_n^{(2)}, \ldots, A_n^{(M)}\right\}$. In the CSMCAS method, by sampling the ancestor index of the conditional particle based on different weights than resampling, retrospective samples of latent variables are realized.

Algorithm 4 Conditional sequential Monte Carlo with ancestor sampling (CSMCAS) method

```
set the previous sample $\boldsymbol{z}_1[k-1]$ as $M$-th particle $\hat{\boldsymbol{z}}_1^{(M)}$ with Eq. (3.1)
generate particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \dots, \hat{\boldsymbol{z}}_1^{(M-1)}\right\}$ with Eq. (2.6)
calculate weights $\left\{w_1^{(1)}, w_1^{(2)}, \dots, w_1^{(M)}\right\}$ with Eq. (2.7)
calculate normalized weights $\left\{W_1^{(1)}, W_1^{(2)}, \dots, W_1^{(M)}\right\}$ with Eq. (2.8)
for $n = 2, \dots, N$ do
    sample ancestor indices $\left\{A_{n-1}^{(1)}, A_{n-1}^{(2)}, \dots, A_{n-1}^{(M-1)}\right\}$ according to the normalized weights $\left\{W_{n-1}^{(1)}, W_{n-1}^{(2)}, \dots, W_{n-1}^{(M)}\right\}$
    calculate backward weights $\left\{\hat{w}_{n-1}^{(1)}, \hat{w}_{n-1}^{(2)}, \dots, \hat{w}_{n-1}^{(M)}\right\}$ with Eq. (3.2)
    calculate normalized backward weights $\left\{\hat{W}_{n-1}^{(1)}, \hat{W}_{n-1}^{(2)}, \dots, \hat{W}_{n-1}^{(M)}\right\}$ with Eq. (3.3)
    sample $M$-th ancestor index $A_{n-1}^{(M)}$ according to the normalized backward weights $\left\{\hat{W}_{n-1}^{(1)}, \hat{W}_{n-1}^{(2)}, \dots, \hat{W}_{n-1}^{(M)}\right\}$
    set the previous sample $\boldsymbol{z}_n[k-1]$ as $M$-th particle $\hat{\boldsymbol{z}}_n^{(M)}$ with Eq. (3.1)
    update particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \dots, \hat{\boldsymbol{z}}_n^{(M-1)}\right\}$ with Eq. (3.4)
    calculate weights $\left\{w_n^{(1)}, w_n^{(2)}, \dots, w_n^{(M)}\right\}$ with Eq. (2.10)
    calculate normalized weights $\left\{W_n^{(1)}, W_n^{(2)}, \dots, W_n^{(M)}\right\}$ with Eq. (2.11)
end for
sample one of the particles $\left\{\hat{\boldsymbol{z}}_{1:N}^{(1)}, \hat{\boldsymbol{z}}_{1:N}^{(2)}, \dots, \hat{\boldsymbol{z}}_{1:N}^{(M)}\right\}$ as $k$-th latent variables $\boldsymbol{z}_{1:N}[k]$ according to the normalized weights $\left\{W_N^{(1)}, W_N^{(2)}, \dots, W_N^{(M)}\right\}$
```

We show the algorithm of the CSMCAS method in Algorithm 4. First, when the initial step $n = 1$, we set the previous sample $\boldsymbol{z}_1[k-1]$ as M-th particle $\hat{\boldsymbol{z}}_1^{(M)}$ as follows:

$$\hat{\boldsymbol{z}}_1^{(M)} \leftarrow \boldsymbol{z}_1[k-1]. \tag{3.1}$$

Next, we generate the particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \dots, \hat{\boldsymbol{z}}_1^{(M-1)}\right\}$ with Eq. (2.6) for the particle numbers $i \in \{1, 2, \dots, M-1\}$. Moreover, we calculate the weights of particles $\left\{w_1^{(1)}, w_1^{(2)}, \dots, w_1^{(M)}\right\}$ with Eq. (2.7), and the normalized weights $\left\{W_1^{(1)}, W_1^{(2)}, \dots, W_1^{(M)}\right\}$ with Eq. (2.8) for all the particle numbers $i \in \{1, 2, \dots, M\}$.

When step $n \in \{2, 3, \ldots, N\}$, the ancestor indices $\left\{A_{n-1}^{(1)}, A_{n-1}^{(2)}, \ldots, A_{n-1}^{(M-1)}\right\}$ are sampled according to the normalized weights $\left\{W_{n-1}^{(1)}, W_{n-1}^{(2)}, \ldots, W_{n-1}^{(M)}\right\}$. This process corresponds to resampling process in the SMC method. Moreover, we calculate the backward weights $\left\{\hat{w}_{n-1}^{(1)}, \hat{w}_{n-1}^{(2)}, \ldots, \hat{w}_{n-1}^{(M)}\right\}$ and the normalized backward weights $\left\{\hat{W}_{n-1}^{(1)}, \hat{W}_{n-1}^{(2)}, \ldots, \hat{W}_{n-1}^{(M)}\right\}$ for the ancestor index of the M-th particle as follows:

$$\hat{w}_{n-1}^{(i)} = W_{n-1}^{(i)} f\left(\boldsymbol{z}_n [k-1] \,\middle|\, \hat{\boldsymbol{z}}_{n-1}^{(i)}, \boldsymbol{\theta}_{\mathrm{sys}}\right), \tag{3.2}$$

$$\hat{W}_{n-1}^{(i)} = \frac{\hat{w}_{n-1}^{(i)}}{\sum_{j=1}^{M} \hat{w}_{n-1}^{(j)}}. \tag{3.3}$$

Note that the normalized weights $W_{n-1}^{(i)}$ is different from the normalized backward weights $\hat{W}_{n-1}^{(i)}$ Furthermore, the ancestor index of the M-th particle $A_{n-1}^{(M)}$ is sampled according to the normalized backward weights $\left\{\hat{W}_{n}^{(1)}, \hat{W}_{n}^{(2)}, \ldots, \hat{W}_{n}^{(M)}\right\}$. For the particle numbers $i = 1, 2, \ldots, M-1$, the latent variables $\hat{\boldsymbol{z}}_n^{(i)}$ at the time step n are sampled using the system model as follows:

$$\hat{\boldsymbol{z}}_n^{(i)} \sim f\left(\boldsymbol{z}_n \,\middle|\, \hat{\boldsymbol{z}}_{n-1}^{\left(A_{n-1}^{(i)}\right)}, \boldsymbol{\theta}_{\mathrm{sys}}\right). \tag{3.4}$$

On the other hand, for the particle number $i = M$, the latent variable $\hat{\boldsymbol{z}}_n^{(M)}$ at time step n are expressed as follows:

$$\hat{\boldsymbol{z}}_n^{(M)} = \boldsymbol{z}_n [k-1]. \tag{3.5}$$

By merging states of each particle up to time step n, the particles are set to be $\hat{\boldsymbol{z}}_{1:n}^{(i)} \leftarrow \left\{\hat{\boldsymbol{z}}_{1:n-1}^{\left(A_{n-1}^{(i)}\right)}, \hat{\boldsymbol{z}}_n^{(i)}\right\}$ for the particle numbers $i \in \{1, 2, \ldots, M\}$. Finally, as in the case of $n = 1$, the weights are calculated and normalized using the Eqs. (2.10) and (2.11). We iterate these process for the time steps $n \in \{2, 3, \ldots, N\}$, and sample one of the particles $\left\{\hat{\boldsymbol{z}}_{1:N}^{(1)}, \hat{\boldsymbol{z}}_{1:N}^{(2)}, \ldots, \hat{\boldsymbol{z}}_{1:N}^{(M)}\right\}$ as the k-th sample of latent variables $\boldsymbol{z}_{1:N} [k]$ according to the normalized weights $\left\{\hat{W}_{N}^{(1)}, \hat{W}_{N}^{(2)}, \ldots, \hat{W}_{N}^{(M)}\right\}$ at time step

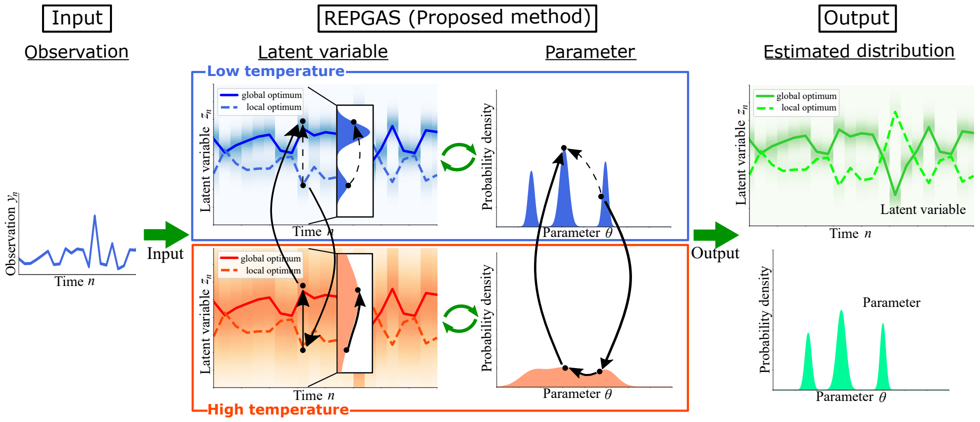

Figure 3.1: Schematic diagrams of the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method. (Left) Observations of time series data $y_{1:N}$ are given as the input. (Middle) The left figures are the distributions of latent variables and the right figures are the distributions of parameters in low and high temperatures. The arrows express the transitions of samples in the REPGAS method. In the REPGAS method, the transitions that are difficult to achieve with the particle-Gibbs with ancestor sampling (PGAS) method (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.

N. The k-th sample of parameters $\boldsymbol{\Theta}\left[k\right]$ is obtained from $p\left(\boldsymbol{\Theta} \mid \boldsymbol{z}_{1:N}\left[k\right], \boldsymbol{y}_{1:N}\right)$ with an MCMC method (e.g., the MH algorithm [8, 38–42]) given the k-th sample of latent variables $\boldsymbol{z}_{1:N}\left[k\right]$.

The PGAS method 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.

3.3 Proposed method: replica exchange particle-Gibbs with ancestor sampling method

In this section, we propose the REPGAS method combining the PGAS method and the replica exchange method for improving the problem influenced by the initial value of the PGAS method. As shown in the schematic diagram of the REPGAS method of Fig. 3.1. 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 the REPGAS method as inputs. In the REPGAS method, we introduce the extension variables called temperature into the PGAS method. By running the PGAS method in multiple temperatures and exchanging samples between temperatures as shown in the middle part of the figure, the transitions that are difficult to achieve with PGAS can be realized by passing through high temperatures. Although Fig. 3.1 shows the case where both the latent variable z_n and the parameter θ are one-dimensional, the REPGAS method is also applicable to the case of multidimensional data.

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

$$\pi_{\mathrm{EX}}\left(\{\boldsymbol{z}_{1:N}\}, \{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right) = \prod_{r=1}^{R} \pi_{T^{(r)}}\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right), \tag{3.6}$$

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

$$\{\boldsymbol{z}_{1:N}\} = \left\{\boldsymbol{z}_{1:N}^{(1)}, \boldsymbol{z}_{1:N}^{(2)}, \ldots, \boldsymbol{z}_{1:N}^{(R)}\right\}, \tag{3.7}$$

$$\{\boldsymbol{\Theta}\} = \left\{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(R)}\right\}. \tag{3.8}$$

Furthermore, the joint posterior distribution at each temperature $\pi_{T^{(r)}}\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$

is represented as follows:

$$\pi_{T^{(r)}}\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right) = \frac{1}{Z\left(T^{(r)}\right)} p\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)^{\frac{1}{T^{(r)}}}, \tag{3.9}$$

where $Z\left(T^{(r)}\right)$ 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 $p\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ is obtained using Bayes' theorem as follows:

$$p\left(\boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right) = \frac{p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{z}_{1:N}^{(r)}, \boldsymbol{\Theta}^{(r)}\right) p\left(\boldsymbol{z}_{1:N}^{(r)} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right) p\left(\boldsymbol{\Theta}^{(r)}\right)}{p\left(\boldsymbol{y}_{1:N}\right)}, \tag{3.10}$$

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

Algorithm 5 Replica exchange particle-Gibbs with ancestor sampling (REPGAS) method

```
initialize the samples of the latent variables $\{\boldsymbol{z}_{1:N}\}[0]$
initialize the samples of the parameters $\{\boldsymbol{\Theta}\}[0]$
for $k = 1, \ldots, K$ do
    for $r = 1, \ldots, R$ do
        sample the latent variables $\boldsymbol{z}_{1:N}^{(r)}[k]$ and the parameters $\boldsymbol{\Theta}^{(r)}[k]$ from the distribution in Eq. (3.9) with one iteration of the PGAS method in Algorithm 3
    end for
    choose replica numbers $r$ and $r+1$ for replica exchange
    calculate $p_{\mathrm{EX}}$ with Eq. (3.11)
    draw an uniform random number $\alpha_{\mathrm{EX}} \sim \mathcal{U}(0,1)$ (where $\mathcal{U}(a,b)$ is an uniform distribution with range $[a,b)$)
    if $\alpha_{\mathrm{EX}} \leq p_{\mathrm{EX}}$ then
        exchange replicas $(\{\boldsymbol{\Theta}\}[k], \{\boldsymbol{z}_{1:N}\}[k]) \leftarrow (\{\boldsymbol{\Theta}^{*}\}[k], \{\boldsymbol{z}_{1:N}^{*}\}[k])$
    end if
end for
```

We show the algorithm of the proposed REPGAS method in Algorithm 5. In the REPGAS method, we obtain samples of the latent variables $\boldsymbol{z}_{1:N}^{(r)}$ and parameters

$\boldsymbol{\Theta}^{(r)}$ for each temperature with the PGAS method according to Eq. (3.9), and exchange samples between temperatures $T^{(r)}$ and $T^{(r+1)}$ according to the following exchange probability:

$$p_{\mathrm{EX}} = \min\left(1, R_{\mathrm{EX}}\right), \tag{3.11}$$

$$R_{\mathrm{EX}} = \frac{\pi_{\mathrm{EX}}\left(\left\{\boldsymbol{z}^{*}_{1:N}\right\}, \left\{\boldsymbol{\Theta}^{*}\right\} \mid \boldsymbol{y}_{1:N}\right)}{\pi_{\mathrm{EX}}\left(\left\{\boldsymbol{z}_{1:N}\right\}, \left\{\boldsymbol{\Theta}\right\} \mid \boldsymbol{y}_{1:N}\right)}, \tag{3.12}$$

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

$$\left\{\boldsymbol{z}^{*}_{1:N}\right\} = \left\{\boldsymbol{z}^{(1)}_{1:N}, \ldots, \boldsymbol{z}^{(r+1)}_{1:N}, \boldsymbol{z}^{(r)}_{1:N}, \ldots, \boldsymbol{z}^{(R)}_{1:N}\right\}, \tag{3.13}$$

$$\left\{\boldsymbol{\Theta}^{*}\right\} = \left\{\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(r+1)}, \boldsymbol{\Theta}^{(r)}, \ldots, \boldsymbol{\Theta}^{(R)}\right\}. \tag{3.14}$$

In the REPGAS method, it becomes possible to overcome the problem of initial value dependence in the PGAS method 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 the PGAS method is able to be run in parallel.

3.4 Experiments

In this section, to verify the effectiveness of the proposed method, we use the following benchmark nonlinear state space model [1,11,17–19]:

$$z_n \sim \mathcal{N}\left(z_n \,\middle|\, \frac{z_{n-1}}{a} + b\frac{z_{n-1}}{1 + z^2_{n-1}} + c\cos\left(dn\right), \sigma^2_z\right), \tag{3.15}$$

$$y_n \sim \mathcal{N}\left(y_n \,\middle|\, \frac{z^2_n}{e}, \sigma^2_y\right), \tag{3.16}$$

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

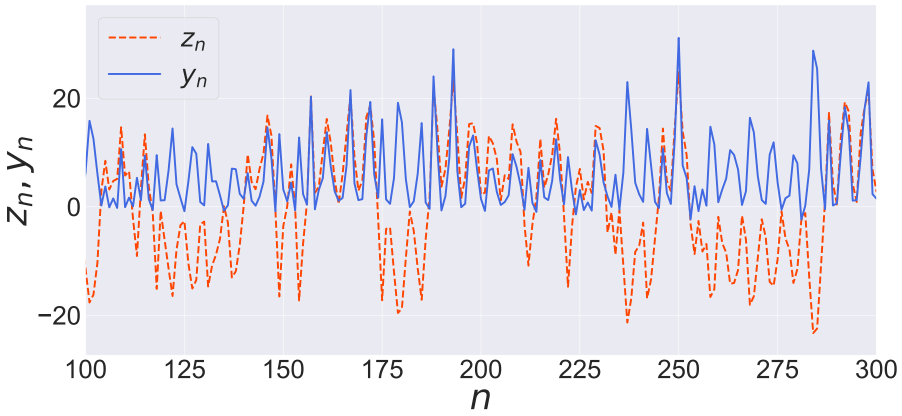

Figure 3.2: The data of the benchmark nonlinear state space model. Observations y_n (blue solid line) and true latent variables z_n (red dashed line) obtained from the benchmark nonlinear state space model at time step $n = 100 - 300$.

variables have 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 $p\left(z_{1:N}, \boldsymbol{\Theta} \mid y_{1:N}\right)$ from the observations $y_{1:N}$ with the parameters $\left\{a, b, c, d, e, \sigma_z^2, \sigma_y^2\right\} = \{2, 25, 8, 1.2, 20, 10, 1\}$ 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. 3.2, 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.

3.4.1 Experiment to compare sampling efficiency

To compare the sampling efficiency of PGAS and the REPGAS method, we estimate the joint posterior distribution of latent variables and parameters $p\left(z_{1:N}, \boldsymbol{\Theta} \mid y_{1:N}\right)$ 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 the REPGAS method is $R = 90$.

We show in Fig. 3.3 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 the REPGAS method samples.

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

3.4.2 Experiment to compare dependence on initial values

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

We show the estimated results of the parameters $\boldsymbol{\Theta}$ and latent variables $z_{1:N}$ in Figs. 3.4, 3.5, and 3.6. Figure 3.4 shows the results of parameters $\boldsymbol{\Theta} = [a, b, c, d]$ estimated with the PGAS method, and Fig. 3.5 shows the results estimated with the REPGAS method. In Figs. 3.4 and 3.5, the vertical axis in each graph represents the value of the probability density function and the horizontal axis represents the

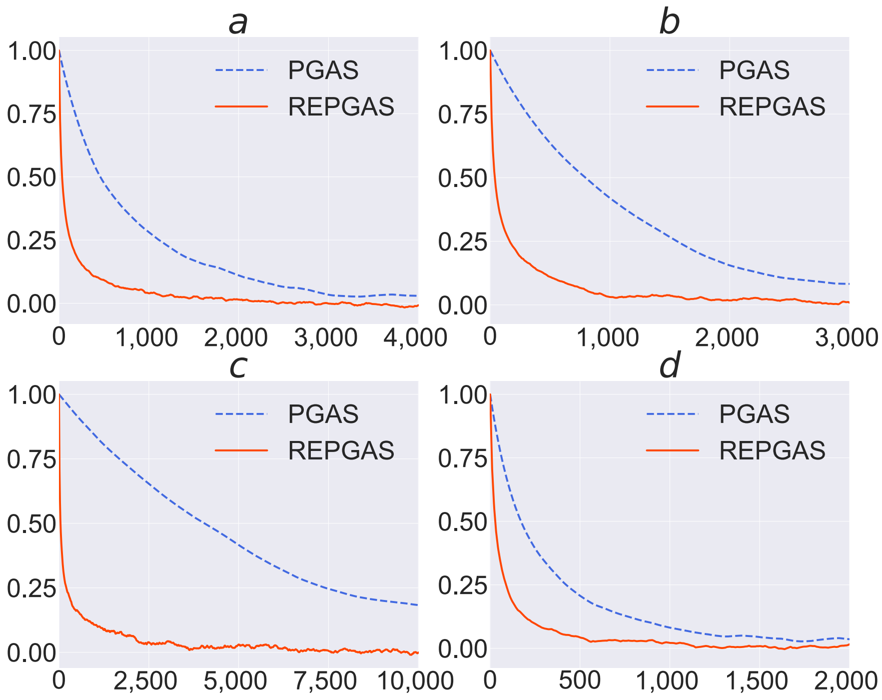

Figure 3.3: Autocorrelation function results of samples obtained by employing the PGAS method and the REPGAS method in the benchmark nonlinear state space model. In each graph, the vertical axis represents the value of the autocorrelation function and the horizontal axis represents the lag length of the autocorrelation function. The blue dashed lines are the results calculated with the PGAS samples of parameters a, b, c, and d, the red solid lines are the results calculated with the REPGAS samples at $T^{(1)} = 1.0$.

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.

As shown in Fig. 3.4, the estimated distributions obtained by the conventional PGAS method are far from the true values. PMCMC methods including PG are guaranteed that the sample from the target distribution can be realized with infinite

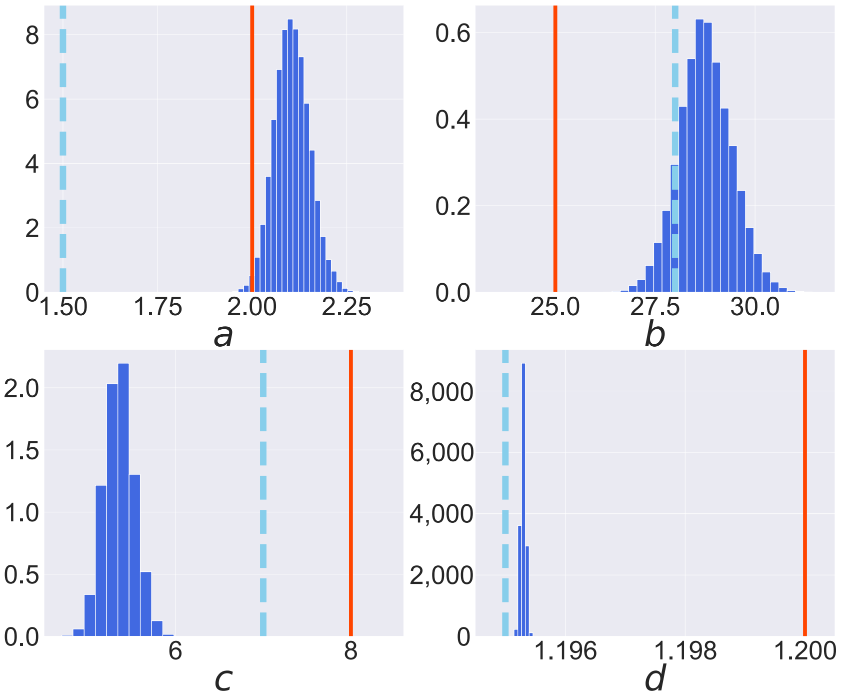

Figure 3.4: Estimated posterior distributions of parameters obtained by employing the PGAS method in the benchmark nonlinear state space model. In each graph the vertical axis represents the value of the probability density function and the horizontal axis represents the values of parameters a, b, c, and d. The red solid lines are true values; the light blue dashed lines are initial values.

samples. However, as shown in Fig. 3.4, 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. 3.5, the peak values of the estimated distributions obtained using the proposed REPGAS method match the true values. Furthermore, the distributions by the REPGAS method have multiple peaks in addition to the true ones, and it can be seen from Fig. 3.5 that the PGAS method is stuck on the peak that is not true.

Figure 3.6 shows the estimated results of the latent variables at time steps be-

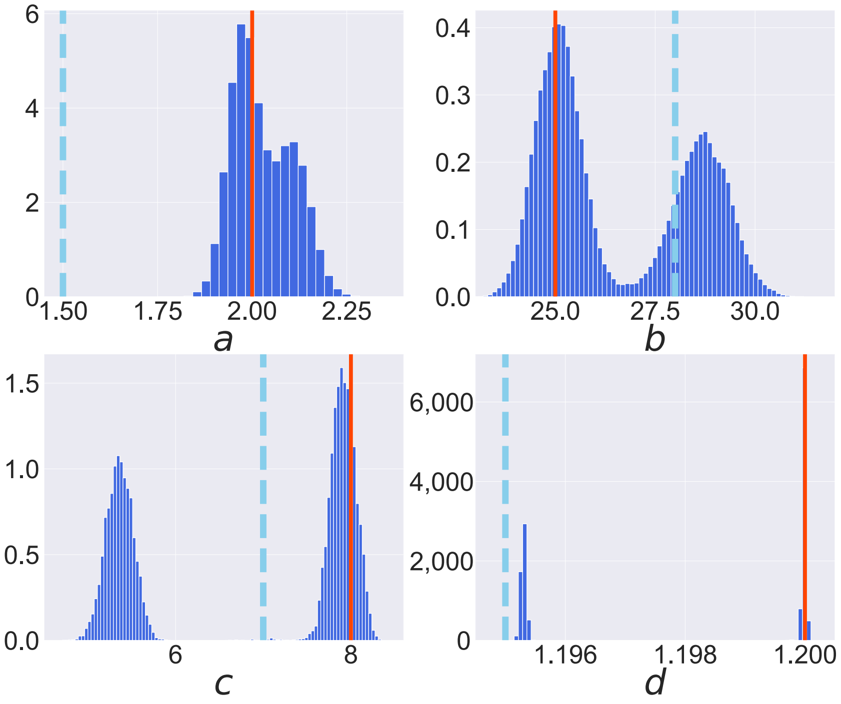

Figure 3.5: Estimated posterior distributions of parameters obtained by employing the REPGAS method in the benchmark nonlinear state space model. See also the captions of the figure and subfigures for Fig 3.4.

tween $n = 100$ and 300, and the upper graph is the result obtained with the PGAS method and the lower graph is the result obtained with the REPGAS method. 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. 3.6, in the estimated distributions obtained using the PGAS method, there are parts where the positive and negative signs of the latent variable z_n are wrong. In consideration of the form of Eq. (3.16), it is considered that there is

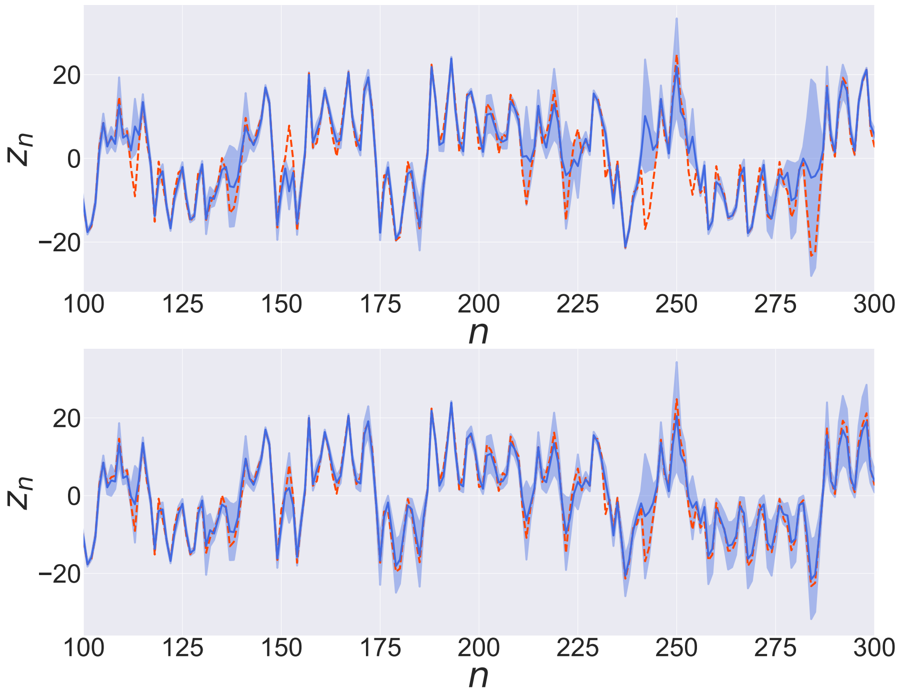

Figure 3.6: Estimated posterior distributions of latent variables obtained by employing the PGAS method and the REPGAS method in the benchmark nonlinear state space model. The upper figure is the result of PGAS and the lower figure is the result of REPGAS. The red dashed lines are true values, the blue 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).

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. 3.5, the estimated distributions of latent variables obtained using the proposed REPGAS method captured the true values.

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

3.5 Summary

In this chapter, we have proposed the REPGAS method, 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 the REPGAS method is improved compared to the PGAS method by combining the PGAS method with the replica exchange method. We have shown in experiments using the benchmark nonlinear state space model that the REPGAS method can improve the problem of initial value dependence of the PGAS method and estimate the joint posterior distribution. In addition to improving the problem, the REPGAS method 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 the REPGAS method compared to the PGAS method.

Chapter 4

Estimation of neural dynamics from spike-train data by employing REPGAS method

4.1 Introduction

Elucidating neuronal dynamics is one of the important subjects to reveal information processing in neural systems. For this purpose, it is necessary to establish a data-driven method for estimating neuronal dynamics from spike-train data. Each neuron in neural systems is known to have various electrical characteristics, such as tonic, phasic, and rebound spikings, and many kinds of neuronal models have been proposed for expressing a part of such electrical responses of the neurons [52–61]. The Izhikevich neuron model, one of neuronal models, is known to reproduce various kinds of electrical responses with low computational cost [60, 61]. However, it is difficult to determine the model parameters that reproduce the electrical responses of neurons since the latent variables of the neurons such as membrane potential and channel variables cannot be observed directly and only one of multidimensional latent variables of neurons or only spike-train data can be observed through partial observations.

In order to estimate neuronal dynamics based on data-driven approaches, methods based on the maximum likelihood (ML) method has been proposed [9,10,12,15]. However, previous studies based on the ML method [9, 10, 12, 15] assumed that we can observe a continuous latent variable such as membrane potential or florescence of calcium imaging rather than spike-train. A previous study proposed an ML method for estimating leaky integrate-and-fire neuron based on spike-train data [62]. However, the kind of neuronal response is limited to tonic spiking and estimated results would be a local optimum. Moreover, other study proposed a method for estimating the Hodgkin-Huxley model [52] based on spike-train data [13]. In this method, the process of obtaining spikes is formulated as the state space model [1–12,14–21,23–26] based on the Hodgkin-Huxley neuron model, and the membrane potentials, which are the latent variables of the neuron, are estimated from spike-train data by using the sequential Monte Carlo (SMC) method [3–5, 8–12, 15–20, 25, 26]. The parameters in the model are also estimated simultaneously by the SMC method with the self-organizing state space model (SOSSM) [5, 13], that the parameters are assumed as the latent variables that do not change over time. However, this method requires that the range of possible values of the parameters be known in advance. Moreover, the accuracy of this method is strongly affected by the problem of the degeneracy in the SMC method.

A method that combines the SMC method and the expectation-maximization (EM) algorithm [8, 22, 28–31] has been proposed for estimating the parameters of state space models [9,10,12,15,20]. This method is a method of sequentially updating the parameters so that the likelihood increases, and it is guaranteed that a local optimum can be estimated. However, this method is highly dependent on the initial values of the parameters used in the estimation, and there is a possibility that a global optimum cannot be estimated.

In this chapter, we propose a data-driven method for estimating the joint posterior distribution of latent variables and parameters in the Izhikevich neuron model from only spike-train data by employing the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method as described in Chapter 3. We assume that only spike-train data is observable due to partial observations, while most of previous works assumed either direct measurements of membrane potentials or imaging measurements of membrane responses [9, 10, 12, 15]. We realize the estimation of the global optimum for the Izhikevich neuron model that expresses various neuronal responses, and realize the precise estimation of neural dynamics from the spike-train data. We also describe the Izhikevich neuron model, formulate the state-space model based on the Izhikevich neuron model, and propose a method for estimating the joint posterior distribution of latent variables and parameters. Moreover, in order to verify the effectiveness of the proposed method, we conduct numerical experiments by using simulated data generated from the Izhikevich neuron model.

4.2 Izhikevich neuron model

The Izhikevich neuron model is one of the neuronal models that represent the membrane potential activities of neurons [60, 61]. Although it is formulated by two relatively simple differential equations for the membrane potential and the membrane recovery variable, it can represent various responses depending on the parameters, and its computational cost is low.

When the membrane potential is expressed as v and the membrane recovery variable is expressed as u, the Izhikevich neuron model is represented by the following two differential equations:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I_{\text{ext}} + \xi_v(t), \tag{4.1}$$

$$\frac{du}{dt} = a(bv - u) + \xi_u(t), \tag{4.2}$$

where I_{ext} is the external input, and a and b are the parameters. In Eqs. (4.1) and (4.2), we consider additive white Gaussian noise terms $\xi_v(t)$ and $\xi_u(t)$ for the Izhikevich neuron model ($\langle \xi_v(t) \rangle = \langle \xi_u(t) \rangle = 0$, $\langle \xi_v(t)\xi_v(s) \rangle = \sigma_v^2 \delta(t-s)$, $\langle \xi_u(t)\xi_u(s) \rangle = \sigma_u^2 \delta(t-s)$ and $\langle \xi_v(t)\xi_u(s) \rangle = 0$, where $\delta(t)$ is the Dirac delta function). Here, standard deviations of membrane potential and membrane recovery variable are expressed by σ_v and σ_u, respectively. As shown in Fig. 4.2, the membrane potential v is affected by the membrane recovery variable u, and the increase of the membrane potential v is suppressed by the increase of the membrane recovery variable u. Moreover, when the membrane recovery variable u decreases, the membrane potential v tends to increase. When the membrane potential v exceeds the threshold value $V_{\text{th}} = 30$, the membrane potential v and the membrane recovery variable u are reset to c and $u + d$, respectively, as follows:

$$v \leftarrow c, \tag{4.3}$$

$$u \leftarrow u + d, \tag{4.4}$$

where c and d are also the parameters.

As described above, there are four parameters $\boldsymbol{\theta} = [a, b, c, d]$ in the Izhikevich neuron model, and various responses can be expressed by adjusting these parameters. In Fig. 4.1 we show the examples of the responses that can be represented by the Izhikevich neuron model. By using different parameters, the various responses, such as the tonic spiking, the phasic spiking, and the mixed mode, are obtained respectively.

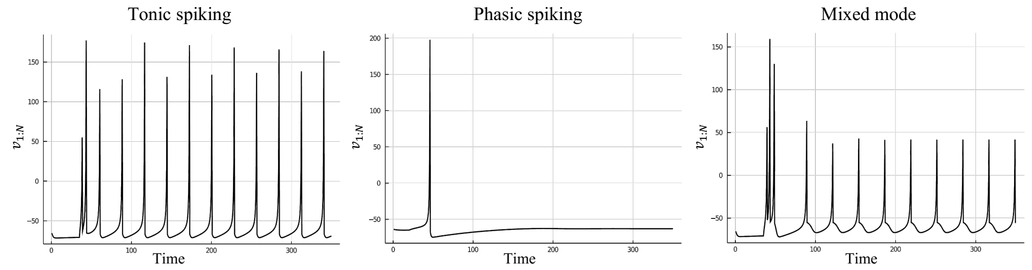

Figure 4.1: Examples of the responses of the Izhikevich neuron model. In the Izhikevich neuron model, various responses can be represented by adjusting parameters $\boldsymbol{\theta} = [a, b, c, d]$. For example, as examples of three typical responses, tonic spiking (left), phasic spiking (center), and mixed mode (right) are reproduced by $\boldsymbol{\theta} = [0.02, 0.20, -65.0, 6.0]$, $\boldsymbol{\theta} = [0.02, 0.25, -65.0, 6.0]$, and $\boldsymbol{\theta} = [0.02, 0.20, -55.0, 4.0]$, respectively.

4.3 Proposed method: replica exchange particle-Gibbs with ancestor sampling method for Izhikevich neuron model

The conceptual diagram of the proposed method is shown in Fig. 4.2. As shown in Fig. 4.2, we propose a method that simultaneously estimates the parameters and the latent variables consisting of the membrane potentials and the membrane recovery variables in the Izhikevich neuron model from spike-train data. In this section, we first explain the Izhikevich neuron model that represents the membrane potential activity of neurons. Moreover, we formulate the process of obtaining spikes as the state-space model based on the Izhikevich neuron model, and propose a data-driven method based on the REPGAS method for the Izhikevich neuron model to simultaneously estimate the parameters and the latent variables such as the membrane potentials and the membrane recovery variables.

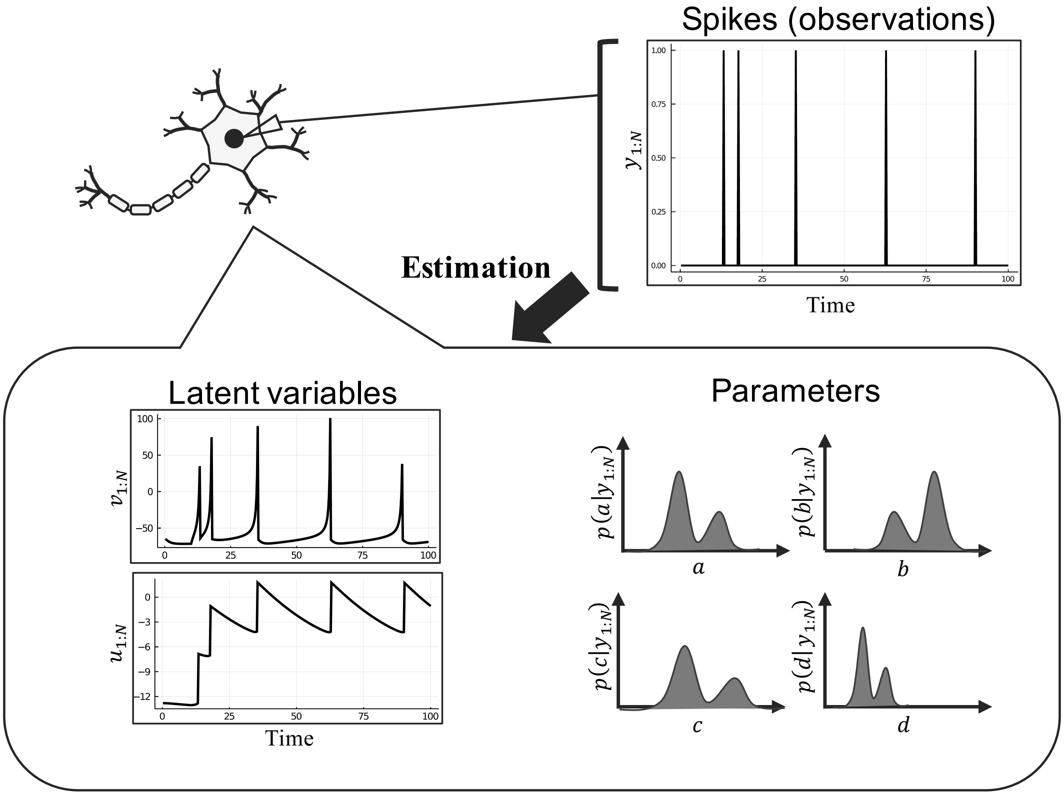

Figure 4.2: Conceptual diagrams of estimating neural dynamics from spike-train data. We propose a method for simultaneously estimating the parameters and the latent variables consisting of the membrane potential $v_{1:N}$ and the membrane recovery variable $u_{1:N}$ in the Izhikevich neuron model from the obtained spikes $y_{1:N}$.

4.3.1 State space model

In this chapter, in order to consider generative process of spike-train data from neuronal dynamics, we discretize the differential equations of the Izhikevich neuron model [Eqs. (4.1)–(4.4)] with respect to time, and consider two kinds of descretized latent variables: the membrane potential v_n and the membrane recovery variable u_n at time step n. Moreover, we consider that the observation is the number of spikes y_n at each time step n.

The system model $p\left(v_n, u_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right)$, a probabilistic model for latent variables v_n, u_n at time step n, is represented as the product of probabilistic models for

v_n and u_n given latent variables at the preceding step and parameters $\boldsymbol{\theta} = [a, b, c, d]$ as follows:

$$p\left(v_n, u_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right) = p\left(v_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right) p\left(u_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right). \tag{4.5}$$

Here, $p\left(v_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right)$ can be derived by employing the Euler-Maruyama method for Eqs. (4.1) and (4.3):

$$p\left(v_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right)$$

$$= \begin{cases} \mathcal{N}\left(v_n \mid v_{n-1} + \Delta_t(0.04v_{n-1}^2 + 5v_{n-1} + 140 - u_{n-1} + I_{\text{ext},n}), \Delta_t\sigma_v^2\right) & v_{n-1} \leq V_{\text{th}} \quad (4.6) \\ \mathcal{N}\left(v_n \mid c + \Delta_t(0.04c^2 + 5c + 140 - u_{n-1} - d + I_{\text{ext},n}), \Delta_t\sigma_v^2\right) & v_{n-1} > V_{\text{th}}, \quad (4.7) \end{cases}$$

where $\mathcal{N}\left(x \mid \mu, \sigma^2\right)$ is the Gaussian distribution with mean μ and variance σ^2, and Δ_t is the finite length between time steps. Until the membrane potential v_{n-1} exceeds the threshold value V_{th}, the membrane potential v_n obeys the distribution with preceding membrane potential v_{n-1} and other terms expressed by Eq. (4.6). When it exceeds the threshold value V_{th}, the membrane potential v_n obeys the distribution with reset membrane potential c and other terms expressed by Eq. (4.7). Similarly, $p\left(u_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right)$ can be derived based on Eqs. (4.2) and (4.4) as follows:

$$p\left(u_n \mid v_{n-1}, u_{n-1}, \boldsymbol{\theta}\right) = \begin{cases} \mathcal{N}\left(u_n \mid u_{n-1} + \Delta_t a(bv_{n-1} - u_{n-1}), \Delta_t\sigma_u^2\right) & v_{n-1} \leq V_{\text{th}} \quad (4.8) \\ \mathcal{N}\left(u_n \mid u_{n-1} + d + \Delta_t a(bc - u_{n-1} - d), \Delta_t\sigma_u^2\right) & v_{n-1} > V_{\text{th}}. \quad (4.9) \end{cases}$$

Furthermore, we assume that the observation model $p\left(y_n \mid v_{1:n+k}\right)$, which is the distribution of the number of spikes y_n for a given set of the membrane potentials, $v_{1:n+k}$, in the time steps from 1 to $n + k$, is expressed by the following Poisson distribution [13]:

$$p\left(y_n \mid v_{1:n+k}\right) = \frac{1}{y_n!} \exp\left(y_n \log\left(\Delta_t\lambda\left(v_{1:n+k}\right)\right) - \Delta_t\lambda\left(v_{1:n+k}\right)\right), \tag{4.10}$$

where $\lambda\left(v_{1:n+k}\right)$ is a function that increases with spiking behavior seen in the membrane potential v around the time step n. We assume that the time interval of Δ_t is sufficiently small and the number of spikes at time step n is either zero or one. The function $\lambda\left(v_{1:n+k}\right)$ is assumed to be obtained using a sigmoid function with temporal convolution as follows:

$$\lambda\left(v_{1:n+k}\right)=\eta\sum_{\tau=1}^{n+k}g\left(v_{\tau}\right)f\left(\tau-\Delta_{t}n\right), \tag{4.11}$$

$$g\left(v\right)=\frac{1}{1+\exp\left(-\beta\left(v-V_{g}\right)\right)}, \tag{4.12}$$

$$f\left(\tau\right)=\begin{cases}p^{-\tau} & \tau\leq 0\\ q^{\tau} & \tau>0\end{cases}, \tag{4.13}$$

where V_g is a constant representing a firing threshold, and η, β, p, and q are positive constants.

4.3.2 Replica exchange particle-Gibbs with ancestor sampling method

We propose a method for simultaneously estimating the membrane potentials $v_{1:N}$, the membrane recovery variables $u_{1:N}$, and the parameters $\boldsymbol{\theta}=[a,b,c,d]$ in the Izhikevich neuron model from spikes $y_{1:N}$ by employing the REPGAS method as described in Chapter 3. The REPGAS method is one of the PMCMC methods, and can be used to obtain samples from the joint posterior distribution $p\left(v_{1:N},u_{1:N},\boldsymbol{\theta}\mid y_{1:N}\right)$ of the latent variables, $v_{1:N}$ and $u_{1:N}$, and the parameters $\boldsymbol{\theta}$. In this chapter, by collecting the samples obtained by the REPGAS method, we estimate the joint posterior distribution $p\left(v_{1:N},u_{1:N},\boldsymbol{\theta}\mid y_{1:N}\right)$.

In the REPGAS method, we introduce R different temperatures $\boldsymbol{T}=\left[T^{(1)},T^{(2)},\ldots,T^{(R)}\right]$ as extension variables and consider the following extended joint posterior distribution:

$$\pi_{\mathrm{EX}}\left(\{v_{1:N}\},\{u_{1:N}\},\{\boldsymbol{\theta}\}\mid y_{1:N}\right)=\prod_{r=1}^{R}\pi_{T^{(r)}}\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right),\tag{4.14}$$

where the membrane potentials $\{v_{1:N}\}$, the membrane recovery variables $\{u_{1:N}\}$, and the parameters $\{\boldsymbol{\theta}\}$ in all temperatures $\boldsymbol{T}$ are represented as $\{v_{1:N}\}=\left\{v_{1:N}^{(1)},\ldots,v_{1:N}^{(R)}\right\}$, $\{u_{1:N}\}=\left\{u_{1:N}^{(1)},\ldots,u_{1:N}^{(R)}\right\}$, and $\{\boldsymbol{\theta}\}=\left\{\boldsymbol{\theta}^{(1)},\ldots,\boldsymbol{\theta}^{(R)}\right\}$, respectively. Furthermore, the joint posterior distribution at each temperature $\pi_{T^{(r)}}\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right)$ is represented as follows:

$$\pi_{T^{(r)}}\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right)=\frac{1}{Z\left(T^{(r)}\right)}p\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right)^{\frac{1}{T^{(r)}}}\tag{4.15}$$

where $Z\left(T^{(r)}\right)$ represents the partition function, and $p\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right)$ is represented as follows:

$$p\left(v_{1:N}^{(r)},u_{1:N}^{(r)},\boldsymbol{\theta}^{(r)}\;\middle|\;y_{1:N}\right)=\frac{p\left(y_{1:N}\;\middle|\;v_{1:N}^{(r)}\right)p\left(v_{1:N}^{(r)},u_{1:N}^{(r)}\;\middle|\;\boldsymbol{\theta}^{(r)}\right)p\left(\boldsymbol{\theta}^{(r)}\right)}{p\left(y_{1:N}\right)},\tag{4.16}$$

where $p\left(y_{1:N}\;\middle|\;v_{1:N}^{(r)}\right)$ and $p\left(v_{1:N}^{(r)},u_{1:N}^{(r)}\;\middle|\;\boldsymbol{\theta}^{(r)}\right)$ are evaluated by using the observation model [Eq. (4.10)] and the system model [Eq. (4.5)] for all time steps, respectively. Here, $p\left(\boldsymbol{\theta}^{(r)}\right)$ is the prior distribution of the parameters $\boldsymbol{\theta}^{(r)}$, and $p\left(y_{1:N}\right)$ is the marginal likelihood.

We obtain samples of the parameters $\boldsymbol{\theta}^{(r)}$ and the latent variables consisting of the membrane potentials $v_{1:N}^{(r)}$ and the membrane recovery variables $u_{1:N}^{(r)}$ alternately for each temperature according to Eq. (4.15). The l-th samples of latent variables, $v_{1:N}^{(r)}[l]$ and $u_{1:N}^{(r)}[l]$, are obtained from $p\left(v_{1:N}^{(r)},u_{1:N}^{(r)}\;\middle|\;v_{1:N}^{(r)}[l-1],u_{1:N}^{(r)}[l-1],y_{1:N},\boldsymbol{\theta}^{(r)}[l-1]\right)$ with the conditional SMC with ancestor sampling. In the conditional SMC with ancestor sampling, the distribution of latent variables is approximated by particles,

$\left\{\hat{v}_{1:N}^{(1)}, \hat{v}_{1:N}^{(2)}, \ldots, \hat{v}_{1:N}^{(M)}\right\}$ and $\left\{\hat{u}_{1:N}^{(1)}, \hat{u}_{1:N}^{(2)}, \ldots, \hat{u}_{1:N}^{(M)}\right\}$, as follows:

$$\begin{aligned} p\left(v_{1:N}, u_{1:N} \,\middle|\, v_{1:N}^{(r)}[l-1], u_{1:N}^{(r)}[l-1], y_{1:N}, \boldsymbol{\theta}^{(r)}[l-1]\right) & \\ \simeq \frac{1}{M}\sum_{i=1}^{M} \delta\left(v_{1:N} - \hat{v}_{1:N}^{(i)}\right)\delta\left(u_{1:N} - \hat{u}_{1:N}^{(i)}\right), & \qquad (4.17) \end{aligned}$$

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

$$W_{n-1}^{(i)} = \frac{w_{n-1}^{(i)}}{\sum_{j=1}^{M} w_{n-1}^{(j)}}, \qquad (4.18)$$

$$w_{n-1}^{(i)} = p\left(y_{n-1} \,\middle|\, \hat{v}_{1:n+k-1}^{(i)}\right), \qquad (4.19)$$

where $w_{n-1}^{(i)}$ is the unnormalized weight for i-th particle. Here, the likelihood of the i-th particle, $p\left(y_{n-1} \,\middle|\, \hat{v}_{1:n+k-1}^{(i)}\right)$, is calculated using the observation model [Eq. (4.10)] with observation data y_{n-1} and membrane potential of i-th particle $\hat{v}_{1:n+k-1}^{(i)}$. Latent variables at the time step n, v_n and u_n, are sampled from the system model $p\left(v_n, u_n \,\middle|\, \hat{v}_{n-1}^{\left(A_{n-1}^{(i)}\right)}, \hat{u}_{n-1}^{\left(A_{n-1}^{(i)}\right)}, \boldsymbol{\theta}^{(r)}[l-1]\right)$. The particles are set to be $\hat{v}_{1:n}^{(i)} \leftarrow \left\{\hat{v}_{1:n-1}^{\left(A_{n-1}^{(i)}\right)}, \hat{v}_{n}^{(i)}\right\}$ and $\hat{u}_{1:n}^{(i)} \leftarrow \left\{\hat{u}_{1:n-1}^{\left(A_{n-1}^{(i)}\right)}, \hat{u}_{n}^{(i)}\right\}$ for particle numbers $i \in \{1, 2, ..., M-1\}$, while the M-th particle is set to be the previous sample $\hat{v}_{1:n}^{(M)} \leftarrow \left\{\hat{v}_{1:n-1}^{\left(A_{n-1}^{(M)}\right)}, v_{n}^{(r)}[l-1]\right\}$ and $\hat{u}_{1:n}^{(M)} \leftarrow \left\{\hat{u}_{1:n-1}^{\left(A_{n-1}^{(M)}\right)}, u_{n}^{(r)}[l-1]\right\}$. Here, the index of ancestor particle $A_{n-1}^{(M)}$ is sampled based on the normalized weights $\left\{\hat{W}_{n-1}^{(1)}, \hat{W}_{n-1}^{(2)}, \ldots, \hat{W}_{n-1}^{(M)}\right\}$ calculated as follows:

$$\hat{W}_{n-1}^{(i)} = \frac{\hat{w}_{n-1}^{(i)}}{\sum_{j=1}^{M} \hat{w}_{n-1}^{(j)}}, \qquad (4.20)$$

$$\hat{w}_{n-1}^{(i)} = W_{n-1}^{(i)} p\left(v_{n}^{(r)}[l-1], u_{n}^{(r)}[l-1] \,\middle|\, \hat{v}_{n-1}^{(i)}, \hat{u}_{n-1}^{(i)}, \boldsymbol{\theta}^{(r)}[l-1]\right), \qquad (4.21)$$

where $\hat{w}_{n-1}^{(i)}$ is the unnormalized weight for sampling the index of ancestor particle $A_{n-1}^{(M)}$. Here, $p\left(v_n^{(r)}\left[l-1\right], u_n^{(r)}\left[l-1\right] \,\middle|\, \hat{v}_{n-1}^{(i)}, \hat{u}_{n-1}^{(i)}, \boldsymbol{\theta}^{(r)}\left[l-1\right]\right)$ is calculated by using the system model [Eq. (4.5)] with the following variables and parameters: the latent variables of previous sample at time step n, $v_n^{(r)}\left[l-1\right]$ and $u_n^{(r)}\left[l-1\right]$, the latent variables of i-th particle at time step $n-1$, $\hat{v}_{n-1}^{(i)}$ and $\hat{u}_{n-1}^{(i)}$, and the parameters of previous sample $\boldsymbol{\theta}^{(r)}\left[l-1\right]$. We iterate the above flow from time step 1 to N and the l-th sample of the latent variables, $v_{1:N}^{(r)}\left[l\right]$ and $u_{1:N}^{(r)}\left[l\right]$, is obtained based on normalized weights $\left\{W_N^{(1)}, W_N^{(2)}, \ldots, W_N^{(M)}\right\}$. The l-th sample of the parameters $\boldsymbol{\theta}^{(r)}\left[l\right]$ is obtained from $p\left(\boldsymbol{\theta} \,\middle|\, v_{1:N}^{(r)}\left[l\right], u_{1:N}^{(r)}\left[l\right], y_{1:N}\right)$ with the Metropolis method [8,38].

In REPGAS, we obtain samples at each temperature as described above, and exchange samples between temperatures $T^{(r)}$ and $T^{(r+1)}$ according to the following exchange probability:

$$p_{\mathrm{EX}} = \min\left(1, R_{\mathrm{EX}}\right), \tag{4.22}$$

$$R_{\mathrm{EX}} = \frac{\pi_{\mathrm{EX}}\left(\left\{v_{1:N}^{*}\right\}, \left\{u_{1:N}^{*}\right\}, \left\{\boldsymbol{\theta}^{*}\right\} \mid y_{1:N}\right)}{\pi_{\mathrm{EX}}\left(\left\{v_{1:N}\right\}, \left\{u_{1:N}\right\}, \left\{\boldsymbol{\theta}\right\} \mid y_{1:N}\right)}, \tag{4.23}$$

where $\{v_{1:N}^{*}\}$, $\{u_{1:N}^{*}\}$, and $\{\boldsymbol{\theta}^{*}\}$ are expressed as follows:

$$\left\{v_{1:N}^{*}\right\} = \left\{v_{1:N}^{(1)}, \ldots, v_{1:N}^{(r+1)}, v_{1:N}^{(r)}, \ldots, v_{1:N}^{(R)}\right\}, \tag{4.24}$$

$$\left\{u_{1:N}^{*}\right\} = \left\{u_{1:N}^{(1)}, \ldots, u_{1:N}^{(r+1)}, u_{1:N}^{(r)}, \ldots, u_{1:N}^{(R)}\right\}, \tag{4.25}$$

$$\left\{\boldsymbol{\theta}^{*}\right\} = \left\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(r+1)}, \boldsymbol{\theta}^{(r)}, \ldots, \boldsymbol{\theta}^{(R)}\right\}. \tag{4.26}$$

We summarize the algorithm of the REPGAS method for Izhikevich neuron model in Algorithm 6. The sampling efficiency of PG is improved by passing through a high temperature state in the replica exchange method. Moreover, it can prevent increasing the calculation time because sampling at each temperature can be parallelized.

Algorithm 6 REPGAS method for Izhikevich neuron model

```
initialize the membrane potentials $\{v_{1:N}\}[0]$ and the membrane recovery variables $\{u_{1:N}\}[0]$
initialize the samples of the parameters $\{\boldsymbol{\theta}\}[0]$
for $l = 1, \ldots, L$ do
    for $r = 1, \ldots, R$ do
        sample the membrane potentials $v_{1:N}^{(r)}[l]$ and the membrane recovery variables $u_{1:N}^{(r)}[l]$ from the distribution in Eq. (4.17) by employing the conditional SMC method
        sample the parameters $\boldsymbol{\theta}^{(r)}[l]$ from the distribution $p\left(\boldsymbol{\theta} \,\middle|\, v_{1:N}^{(r)}[l], u_{1:N}^{(r)}[l], y_{1:N}\right)$ by employing the Metropolis method
    end for
    choose replica numbers $r$ and $r+1$ for replica exchange
    calculate $p_{\mathrm{EX}}$ with Eq. (4.22)
    draw a uniform random number $\alpha_{\mathrm{EX}}$ with range $[0, 1)$
    if $\alpha_{\mathrm{EX}} \leq p_{\mathrm{EX}}$ then
        exchange replicas
        $(\{v_{1:N}\}[l], \{u_{1:N}\}[l], \{\boldsymbol{\theta}\}[l]) \leftarrow (\{v_{1:N}^{*}\}[l], \{u_{1:N}^{*}\}[l], \{\boldsymbol{\theta}^{*}\}[l])$
    end if
end for
```

4.4 Experiments

In this section, we verify the effectiveness of the proposed method. First, we estimate the joint posterior distribution $p(v_{1:N}, u_{1:N}, \boldsymbol{\theta} \mid y_{1:N})$ of the parameters $\boldsymbol{\theta}$ and the latent variables consisting of the membrane potentials $v_{1:N}$ and the membrane recovery variables $u_{1:N}$ by employing the proposed method for the simulation data generated from the Izhikevich neuron model. Next, the estimation results of the proposed method and other methods are compared for some typical response characteristic parameters. Finally, the reproducibility of the response is verified with the inputs not used for estimation of parameters and the parameters estimated by each method.

We use the simulated data generated with the four parameter sets (i) tonic spiking, (ii) phasic spiking, (iii) mixed mode, and (iv) rebound spike shown in Table 4.1. The variances of the membrane potentials $v_{1:N}$ and the membrane recovery

Table 4.1: True parameters of the Izhikevich neuron model for the experiments.

Model	a	b	c	d
(i) Tonic spiking	0.02	0.20	-65.0	6.0
(ii) Phasic spiking	0.02	0.25	-65.0	6.0
(iii) Mixed mode	0.02	0.20	-55.0	4.0
(iv) Rebound spike	0.03	0.25	-60.0	4.0

variables $u_{1:N}$ are $\sigma_v^2 = 0.25$ and $\sigma_u^2 = 10^{-4}$, respectively. As an example of the data used, the data generated with the parameters of (i) tonic spiking is shown in Fig. 4.3, where the true membrane potentials $v_{1:N}$, the true membrane recovery variables $u_{1:N}$, the spikes $y_{1:N}$, and the input currents $I_{\mathrm{ext},1:N}$ are shown in order from the top. The horizontal axes represent the time, and the vertical axes represent the value of each variable. The latent variables, $v_{1:N}$ and $u_{1:N}$, and the parameters $\boldsymbol{\theta}$ are estimated from spikes $y_{1:N}$ obtained by applying the same input currents $I_{\mathrm{ext},1:N}$ to other parameter sets. In the proposed method, the number of particles M is 50, the number of samples L is 3×10^5, the number of burn-in samples $L_{\mathrm{burn-in}}$ is 1.5×10^5, the number of replicas R is 64, and the temperatures $\boldsymbol{T}$ are set as $\boldsymbol{T} = \left[1.0, 1.1, 1.1^2, \ldots, 1.1^{R-1}\right]$. Moreover, we assume that the prior distribution $p(\boldsymbol{\theta})$ is a uniform distribution. In this section, we focus on estimating parameters $\boldsymbol{\theta} = [a, b, c, d]$ and assume that the variances $\left\{\sigma_v^2, \sigma_u^2\right\}$ of the membrane potentials $v_{1:N}$ and the membrane recovery variables $u_{1:N}$ are known. Furthermore, for all estimation results, the parameter set of (i) tonic spiking is used as the initial value, and the hyper parameters for the observation model are $\eta = 1.02$, $\beta = 0.28$, $V_g = -19.5$, $p = 0.23$, and $q = 0.05$.

Figures 4.4 and 4.5 are the estimated distributions of the parameters $\boldsymbol{\theta} = [a, b, c, d]$ of (iii) mixed mode and (iv) rebound spike by using the proposed method. As shown in Fig. 4.4, even though true values of c and d ($c = -55.0$, $d = 4.0$) are far from initial values ($c = -65.0$, $d = 6.0$), the true values of (iii) mixed mode are found to be estimated appropriately. Also in Fig. 4.5, although all the true param-

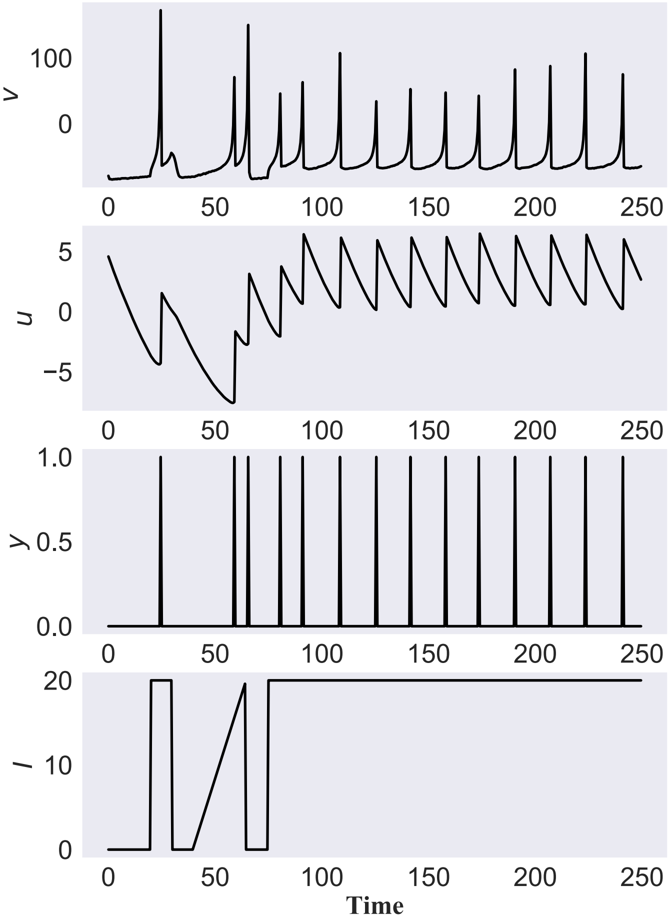

Figure 4.3: Generated data with the parameters of (i) tonic spiking. From the top, they are the graphs of the true membrane potentials $v_{1:N}$, the true membrane recovery variable $u_{1:N}$, the spikes $y_{1:N}$, and the input currents $I_{\mathrm{ext},1:N}$. The horizontal axes show the time and the vertical axes show the value of each variable. All the data are generated by applying the same input currents $I_{\mathrm{ext},1:N}$ for each parameter set.

eters ($a = 0.03$, $b = 0.25$, $c = -60.0$, $d = 4.0$) are different from the initial values ($a = 0.02$, $b = 0.20$, $c = -65.0$, $d = 6.0$), all the parameters show high probability densities around the true values of (iv) rebound spike, and it can be verified that the estimations can be performed appropriately. Moreover, since not only the global optimum but also the local optimum can be found simultaneously, it can be verified that a wide range can be searched. Figures 4.6 and 4.7 are the estimated results of the latent variables such as the membrane potentials $v_{1:N}$ and the membrane recovery variables $u_{1:N}$. There the horizontal axes represent the time, and the vertical axes represent the value of each latent variable. In each graph, the solid lines represent the true values and the filled areas represent the 95% confidence intervals of the estimated distributions. In both figures, the complex response of the membrane potential can be properly captured, and it can be verified that the estimations are properly performed.

Next, we show in the Table 4.2 the estimated results of all parameter sets. As results estimated by other methods, we show the results estimated by the SMC method with the SOSSM, the method to estimate the parameters in the SMC method by considering the parameters as a part of latent variables [5,13], and results estimated by the SMC method with the EM algorithm, a point estimation method that sequentially updates the parameters so that the likelihood increase [9,10,12,15,20]. In the SMC method with the SOSSM, the number of particles M_{SOSSM} is 5×10^5, and the parameters of the initial particles are generated as follows using a uniform distribution:

$$a \sim U\left(0, 0.5\right), \tag{4.27}$$

$$b \sim U\left(-1.5, 1.0\right), \tag{4.28}$$

$$c \sim U\left(-70, -50\right), \tag{4.29}$$

$$d \sim U\left(3, 10\right). \tag{4.30}$$

In the SMC method with the EM algorithm, the number of particles M_{EM} is 10^3, the number of iterations L_{EM} is 10^4, and the initial values of parameters are the

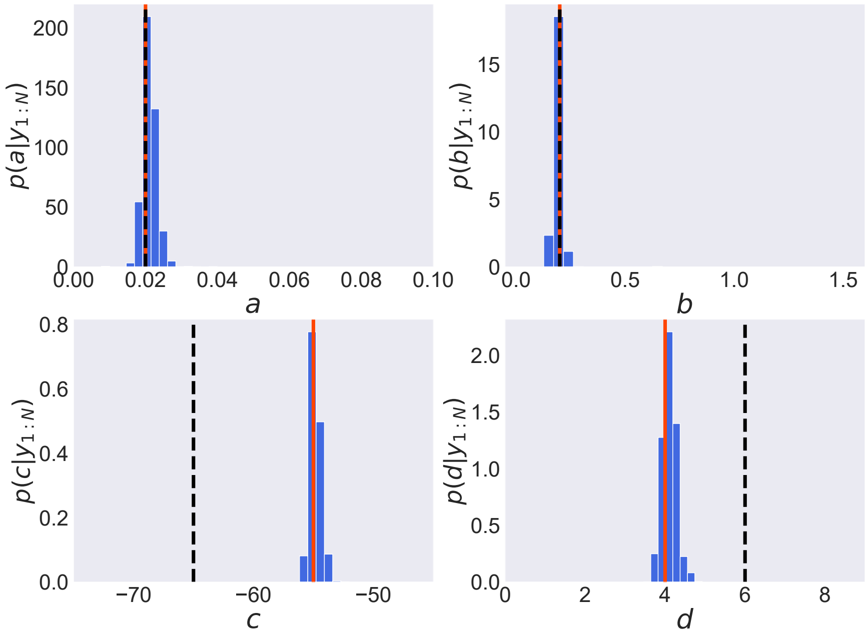

Figure 4.4: Estimated result of parameters of (iii) mixed mode by the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method. The horizontal axes represent the value of each parameter, the vertical axes represent the probability densities, the red solid lines represent the true values, the black dashed lines represent the initial values, and the blue histograms represent the estimated distributions.

parameters of (i) tonic spiking as in the proposed method. In the proposed method and the SMC method with the SOSSM, the mode values of the estimated distributions are used as the estimated results. One can see from the results in Table 4.2, that values close to the true values can be estimated by the SMC method with the SOSSM. However, it is considered that the particles closer to the true values have been lost due to the degeneracy of the SMC method since the standard deviations of the result of (iii) mixed mode are very small values. Regarding the results of the SMC method with the EM algorithm, compared to the result of (i) tonic spiking,

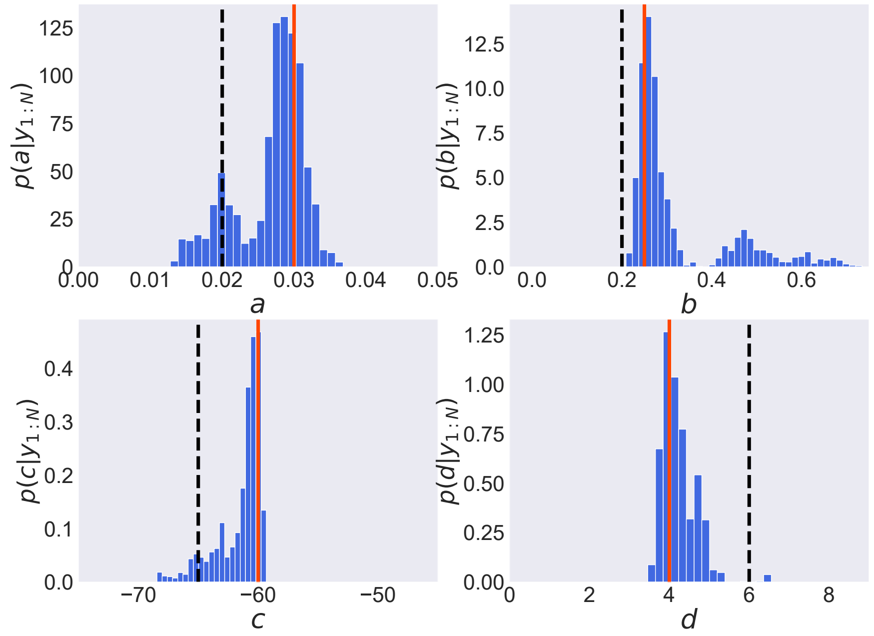

Figure 4.5: Estimated result of parameters of (iv) rebound spike by the REPGAS method. See also the captions of the figure and subfigures for Fig. 4.4.

which starts the estimation from the true values, the estimation results of the other parameter sets are poor. Since this method is a point estimation method, it is considered that the global optimum could not be estimated without getting out of the local optimum. In the proposed method, it can be verified that the estimated values are closest to the true values in all parameter sets even though the proposed method uses the same initial values of parameters as the SMC method with EM algorithm. In other words, the proposed method using a set of initial values achieves more accurate estimation than the SMC method with the SOSSM using many sets of initial values.

Finally, in Figs. 4.8, 4.9, 4.10, and 4.11, we show results when comparing the reproducibilities of the responses to the new inputs with the parameter sets estimated

Table 4.2: Estimated results of parameters in the Izhikevich neuron model.

Model	Parameter	True	SOSSM		EM	REPGAS	
			Mode	Std		Mode	Std
(i) Tonic spiking	a	0.0200	0.0251	0.0081	0.0216	**0.0207**	0.0068
	b	0.200	0.151	0.289	0.181	**0.193**	0.389
	c	−65.00	−60.04	4.12	−63.70	**−64.13**	6.20
	d	6.00	**5.94**	1.15	5.92	**5.94**	1.05
(ii) Phasic spiking	a	0.0200	0.0222	0.0039	0.0255	**0.0202**	0.0058
	b	0.250	0.224	0.168	0.146	**0.250**	0.153
	c	−65.00	−67.10	1.36	−87.46	**−65.39**	2.63
	d	6.00	6.04	0.27	6.08	**6.00**	0.69
(iii) Mixed mode	a	0.0200	0.0170	0.00002	0.0119	**0.0202**	0.0019
	b	0.200	0.222	0.0007	0.336	**0.201**	0.021
	c	−55.00	−55.33	0.01	−73.31	**−54.98**	0.53
	d	4.00	3.62	0.01	5.79	**4.01**	0.20
(iv) Rebound spike	a	0.0300	0.0387	0.0020	0.0154	**0.0299**	0.0048
	b	0.250	0.224	0.011	0.272	**0.252**	0.115
	c	−60.00	−58.72	0.59	−87.50	**−60.00**	1.87
	d	4.00	4.62	0.22	6.00	**4.00**	0.43

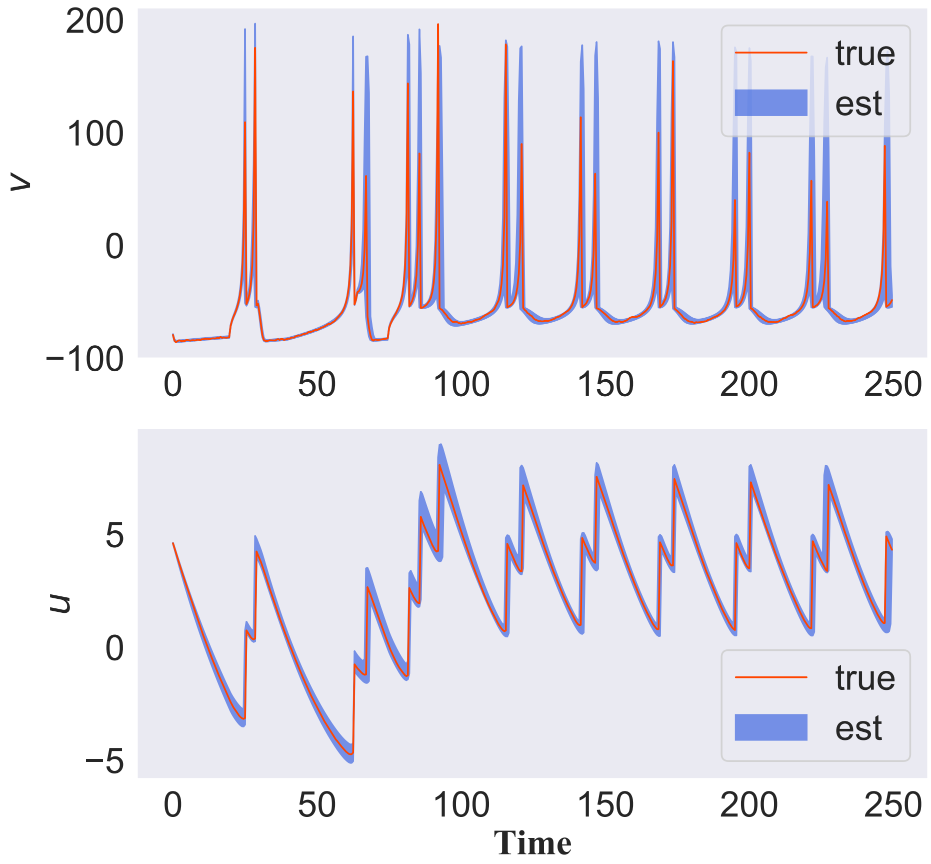

Figure 4.6: Estimated result of latent variables of (iii) mixed mode by the REPGAS method. The horizontal axes represent the time, the vertical axes represent the value of each latent variable, the red solid lines are the true values, and the blue filled areas represent the 95% confidence intervals of the estimated distributions.

by each method for (i) tonic spiking, (ii) phasic spiking, (iii) mixed mode, and (iv) rebound spike, respectively. In each figure, the upper graphs show the reproduction results of the membrane potentials $v_{1:N}$, the lower graphs show the input currents $I_{\mathrm{ext},1:N}$, the vertical axes show the value of each variable, and the horizontal axes show the time. In the upper graphs, the solid lines show the responses obtained

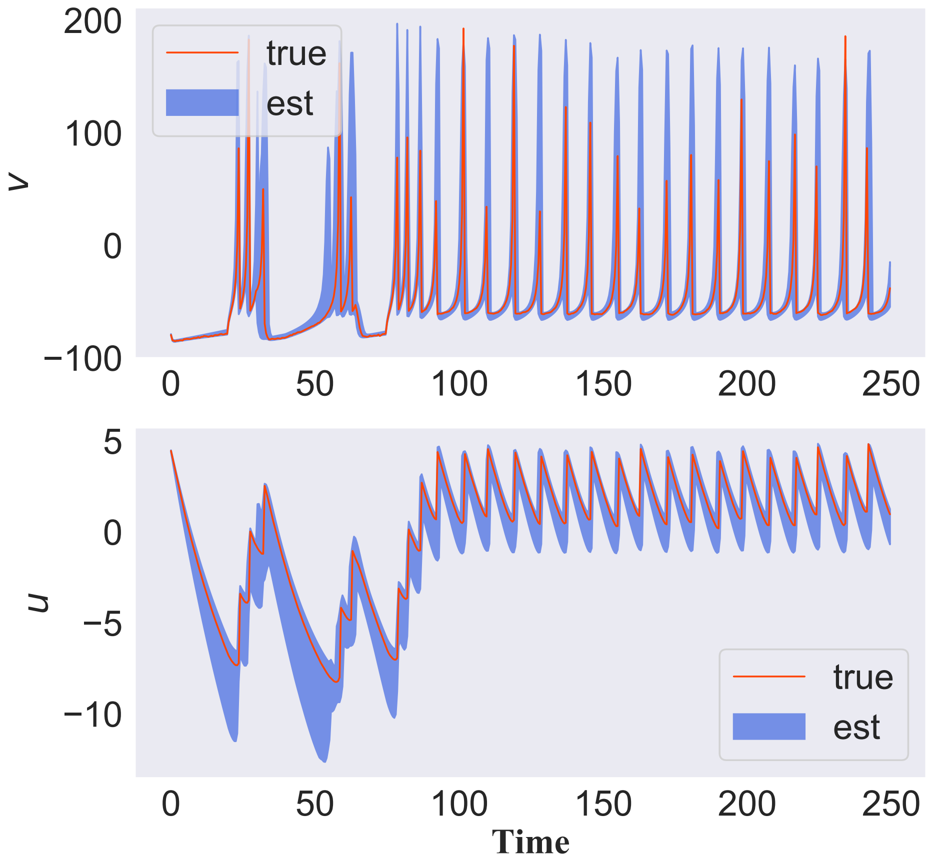

Figure 4.7: Estimated result of latent variables of (iv) rebound spike by the REPGAS method. See also the captions of the figure and subfigures for Fig. 4.6.

when using the true parameters, and the dashed lines show the responses obtained when using the parameters estimated by each method. In all figures, it can be verified that the proposed method can output spikes at timings closer to the simulation results with true parameters than the results of the other methods. It is also verified in Figs. 4.9 and 4.11 that the number of spikes is correct only in the results of the proposed method. Table 4.3 shows the quantitative comparison results of

Table 4.3: Reproducibility comparison results for new inputs.

	Model	True	SOSSM	EM	REPGAS
RMSE	(i) Tonic spiking		25.76	23.74	**18.09**
	(ii) Phasic spiking		7.04	10.21	**0.09**
	(iii) Mixed mode		22.49	24.54	**10.57**
	(iv) Rebound spike		7.43	10.62	**7.11**
Number of spikes	(i) Tonic spiking	5	**5**	**5**	**5**
	(ii) Phasic spiking	1	0	0	**1**
	(iii) Mixed mode	6	8	**6**	**6**
	(iv) Rebound spike	1	0	2	**1**
Average deviation time of the spike timings	(i) Tonic spiking		+8.1	+2.6	**+0.5**
	(ii) Phasic spiking		–	–	**0.0**
	(iii) Mixed mode		−19.67	17.58	**−0.5**
	(iv) Rebound spike		–	−5.0	**−3.0**

each method. There we compare the root mean square error (RMSE), the number of spikes, and the average deviation time of the spike timings when the membrane potentials are reproduced by each parameter. In the average deviation time of the spike timings, "–" is displayed when no spikes can be obtained and the calculation cannot be performed. It can be verified that the REPGAS method shows the best results for all results. The above results confirm that Izhikevich neuron model parameters that can reproduce actual responses can be estimated from the spike-train data by using the proposed method.

4.5 Summary

In this chapter, we proposed the method to estimate the parameter and the latent variables consisting of the membrane potentials and the membrane recovery variables of the Izhikevich neuron model from only spike-train data based on the REPGAS method. In the proposed method, the process of obtaining spikes is formulated as the state space model based on the Izhikevich neuron model, and the latent variables and the parameters are estimated simultaneously by employing the REPGAS

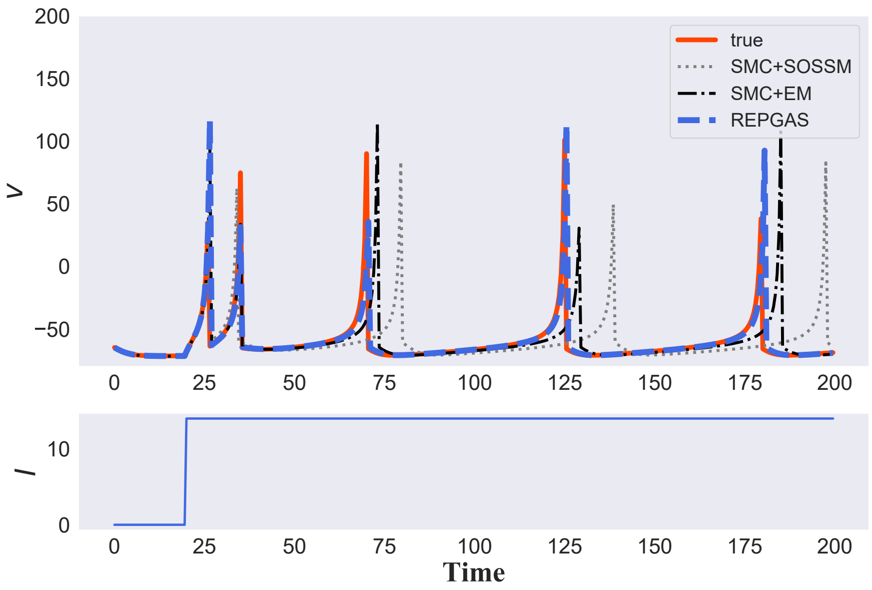

Figure 4.8: Comparison of reproduction results of (i) tonic spiking response. The upper graph shows the membrane potentials $v_{1:N}$, the lower graph shows the input currents $I_{\mathrm{ext},1:N}$, the horizontal axes show the time, and the vertical axes show the value of each variable. In the upper graph, the response obtained when using the true parameters (red solid line), the responses obtained when using the parameters estimated by the REPGAS method (blue dashed line), the responses obtained when using the parameters estimated by the sequential Monte Carlo (SMC) method with the expectation-maximization (EM) algorithm (black dashed line), and the responses obtained when using the parameters estimated by the SMC method with the self-organizing state space model (SOSSM) (gray dashed line) are shown.

method. Moreover, we verified that the latent variables and the parameters of the Izhikevich neuron model can be estimated simultaneously by using the proposed method for simulated data generated from the Izhikevich neuron model with several parameter sets. Furthermore, it was also shown that the proposed method can estimate parameters closer to the true values than the SMC method with the SOSSM and the SMC method with the EM algorithm, and that the responses to new inputs is more reproducible.

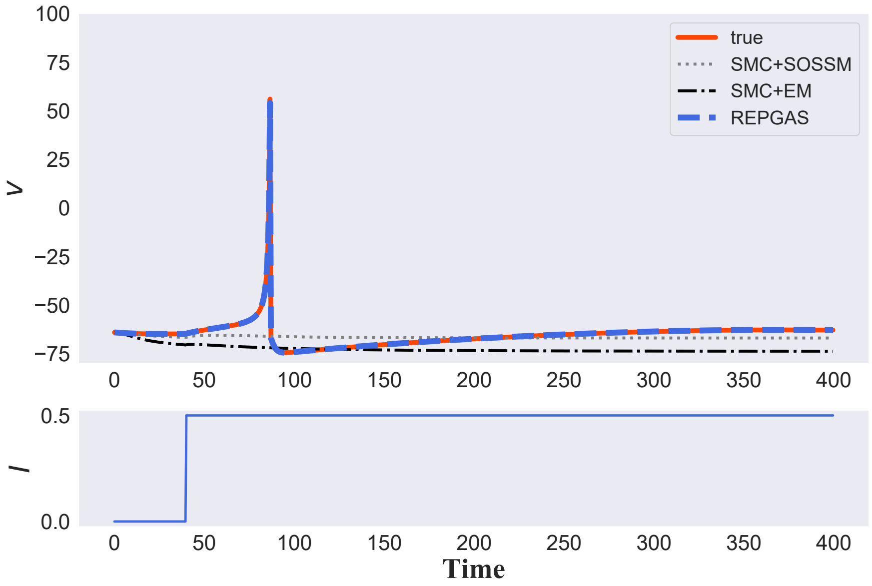

Figure 4.9: Comparison of reproduction results of (ii) phasic spiking response. See also the captions of the figure and subfigures for Fig. 4.8.

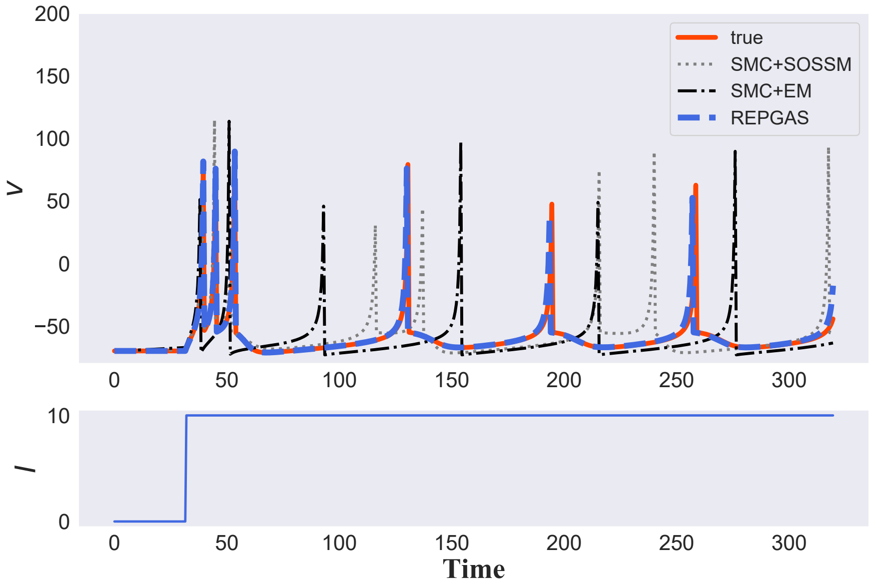

Figure 4.10: Comparison of reproduction results of (iii) mixed mode response. See also the captions of the figure and subfigures for Fig. 4.8.

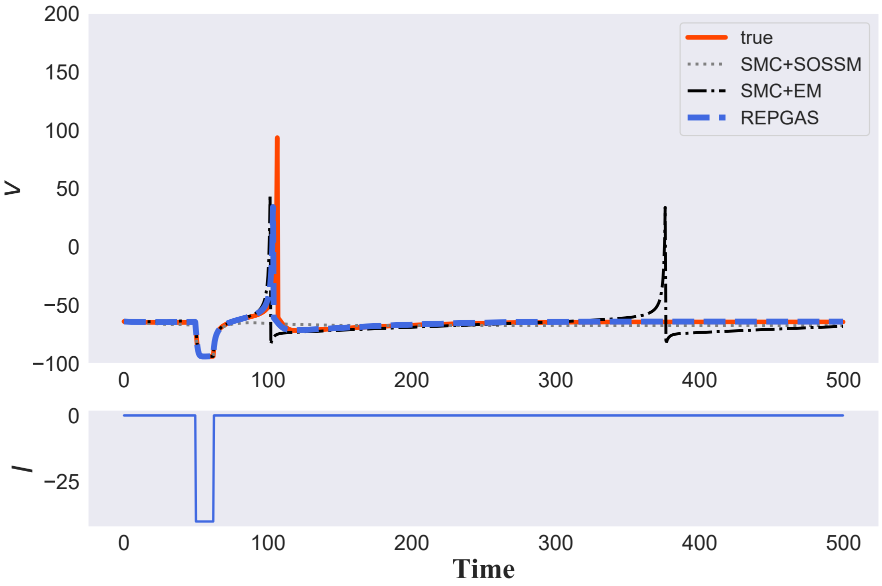

Figure 4.11: Comparison of reproduction results of (iv) rebound spike response. See also the captions of the figure and subfigures for Fig. 4.8.

Chapter 5

Replica exchange particle marginal Metropolis-Hastings method

5.1 Introduction

In order to estimate the distribution of parameters in a state space model, two kinds of the particle Markov chain Monte Carlo (PMCMC) methods have been proposed: the particle-Gibbs (PG) method and the particle marginal Metropolis-Hastings (PMMH) method [11]. Both methods are combining Markov chain Monte Carlo (MCMC) methods with the SMC method, and the distribution of parameters is estimated by collecting samples. The PG method combines the SMC method with Gibbs sampling [8,32–35,37], and the PG method samples latent variables and parameters in a state space model from the joint posterior distribution of latent variables and parameters alternately. In the PG method, the SMC method is employed for sampling latent variables. The PMMH method, on the other hand, combines the SMC method with the Metropolis-Hastings (MH) algorithm [8,38–42], and the PMMH method samples parameters in a state space model from the marginal posterior distribution of parameters. In the PMMH method, the SMC method is employed for calculating the likelihood marginalized over the distribution of latent variables. Both the PG method and the PMMH method have been widely applied (for example, the PG method [43–45], the PMMH method [46–48]).

In Chapter 3, we proposed the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method in order to improve the problem of initial value dependence in the PG method. However, the PMMH method also has the problem of initial value dependence . Therefore, it is important to improve the PMMH method for accurate estimation of parameters.

In this chapter, we propose the replica exchange particle marginal Metropolis-Hastings (REPMMH) method, which combines the PMMH method with the replica exchange method [49–51] in order to improve the problem of initial value dependence in the PMMH method. Combining the replica exchange method with the PMMH method makes it possible to estimate the parameters governing the dynamics for very complex and nonlinear time-series data. We first describe the PMMH method as a conventional method. Then, after explaining the proposed method, we conduct experiments to compare the proposed method with the conventional methods, the PMMH method and the REPGAS method.

5.2 Methods

In this section, we propose the replica exchange particle marginal Metropolis-Hastings (REPMMH) method. First, we describe the conventional particle marginal Metropolis-Hastings (PMMH) method to estimate the marginal posterior distribution of parameters obtained by marginalization over the distribution of latent variables in a state space model. After that, we propose the REPMMH method that combines the PMMH method with the replica exchange method to improve the problem of initial value dependence in the PMMH method.

5.2.1 Conventional method: particle marginal Metropolis-Hastings method

The colorredPMMH method combines the sequential Monte Carlo (SMC) method [3–5, 8–12, 15–20, 25, 26] with the Metropolis-Hastings (MH) algorithm [8, 38–42]. The PMMH method was proposed to estimate the marginal posterior distribution of parameters $p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$ for time-series observations $\boldsymbol{y}_{1:N}$ represented as a state space model [11].

In the PMMH method, the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ is used to evaluate the appropriateness of parameters $\boldsymbol{\Theta}$. Here, the SMC method is used to calculate the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ of the parameters $\boldsymbol{\Theta}$ obtained by marginalization over the distribution of latent variables $\boldsymbol{z}_{1:N}$. A new sample candidate of parameters $\boldsymbol{\Theta}^* = \left\{\boldsymbol{\theta}^*_{\mathrm{sys}}, \boldsymbol{\theta}^*_{\mathrm{obs}}\right\}$ is proposed from an arbitrary proposal distribution $q\left(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}\left[k-1\right]\right)$ given the sample one step before $\boldsymbol{\Theta}\left[k-1\right]$, where k is the sample number. Moreover, whether to accept or reject the sample candidate $\boldsymbol{\Theta}^*$ is determined based on the following acceptance probability:

$$p_{\mathrm{accept}} = \min\left(1, \frac{p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}^*\right) p\left(\boldsymbol{\Theta}^*\right)}{p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\left[k-1\right]\right) p\left(\boldsymbol{\Theta}\left[k-1\right]\right)} \frac{q\left(\boldsymbol{\Theta}\left[k-1\right] \mid \boldsymbol{\Theta}^*\right)}{q\left(\boldsymbol{\Theta}^* \mid \boldsymbol{\Theta}\left[k-1\right]\right)}\right), \tag{5.1}$$

where $p\left(\boldsymbol{\Theta}\right)$ represents the prior distribution of parameters. $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ is the marginal likelihood obtained by marginalization over the distributions of latent variables $\boldsymbol{z}_{1:N}$ as follows:

$$\begin{aligned} p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right) &= \int p\left(\boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N} \mid \boldsymbol{\Theta}\right) d\boldsymbol{z}_{1:N} \\ &= \int p\left(\boldsymbol{y}_1 \mid \boldsymbol{z}_1, \boldsymbol{\theta}_{\mathrm{obs}}\right) p\left(\boldsymbol{z}_1\right) \prod_{n=2}^{N} p\left(\boldsymbol{y}_n \mid \boldsymbol{z}_n, \boldsymbol{\theta}_{\mathrm{obs}}\right) p\left(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{\theta}_{\mathrm{sys}}\right) d\boldsymbol{z}_{1:N}, \end{aligned}$$

where $p\left(\boldsymbol{z}_1\right)$ is the distribution of latent variables $\boldsymbol{z}_1$ at time step 1. Since it is difficult to obtain the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ analytically, the SMC method is used in the PMMH method to calculate the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ numerically.

The SMC method estimates the distribution of latent variables by approximating the distribution with the density of the particles $\left\{\hat{\boldsymbol{z}}_{1:N}^{(1)}, \hat{\boldsymbol{z}}_{1:N}^{(2)}, \ldots, \hat{\boldsymbol{z}}_{1:N}^{(M)}\right\}$ as follows:

$$p\left(\boldsymbol{z}_{1:N} \mid \boldsymbol{y}_{1:N}, \boldsymbol{\Theta}\right) \simeq \frac{1}{M} \sum_{i=1}^{M} \delta\left(\boldsymbol{z}_{1:N} - \hat{\boldsymbol{z}}_{1:N}^{(i)}\right), \tag{5.2}$$

where $\hat{\boldsymbol{z}}_{1:N}^{(i)}$ is the i-th particle and M is the number of particles. $\delta\left(\boldsymbol{z}_{1:N}\right)$ is the Dirac delta distribution.

To obtain particles $\left\{\hat{\boldsymbol{z}}_{n}^{(1)}, \hat{\boldsymbol{z}}_{n}^{(2)}, \ldots, \hat{\boldsymbol{z}}_{n}^{(M)}\right\}$ at a time step n, we sample the i-th particle $\hat{\boldsymbol{z}}_{n}^{(i)}$ at the time step n from the i-th particle $\hat{\boldsymbol{z}}_{n-1}^{(i)}$ at the previous time step $n-1$ for each $i \in \{1, 2, \ldots, M\}$ with the system model as follows:

$$\hat{\boldsymbol{z}}_{n}^{(i)} \sim p\left(\boldsymbol{z}_{n} \,\middle|\, \hat{\boldsymbol{z}}_{n-1}^{(i)}, \boldsymbol{\theta}_{\text{sys}}\right). \tag{5.3}$$

Moreover, the obtained particles $\left\{\hat{\boldsymbol{z}}_{n}^{(1)}, \hat{\boldsymbol{z}}_{n}^{(2)}, \ldots, \hat{\boldsymbol{z}}_{n}^{(M)}\right\}$ are resampled based on the normalized weights $\left\{W_{n}^{(1)}, W_{n}^{(2)}, \ldots, W_{n}^{(M)}\right\}$ obtained as follows:

$$W_{n}^{(i)} = \frac{w_{n}^{(i)}}{\sum_{j=1}^{M} w_{n}^{(j)}}, \tag{5.4}$$

$$w_{n}^{(i)} = p\left(\boldsymbol{y}_{n} \,\middle|\, \hat{\boldsymbol{z}}_{n}^{(i)}, \boldsymbol{\theta}_{\text{obs}}\right). \tag{5.5}$$

By iterating the above flow for time step $n \in \{1, 2, \ldots, N\}$, particles that approximate the distribution of latent variables $\boldsymbol{z}_{1:N}$ can be obtained. Here, the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ can be calculated approximately as follows:

$$p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right) = \prod_{n=1}^{N} p\left(\boldsymbol{y}_{n} \mid \boldsymbol{y}_{1:n-1}, \boldsymbol{\Theta}\right) \simeq \frac{1}{M} \prod_{n=1}^{N} \sum_{i=1}^{M} w_{n}^{(i)}. \tag{5.6}$$

By calculating the acceptance probability p_{accept} in Eq. (5.1) with the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}^{*}\right)$ for the sample candidate $\boldsymbol{\Theta}^{*}$ obtained by Eq. (5.6), it is

Algorithm 7 Particle marginal Metropolis-Hastings (PMMH) method

```
initialize the parameters $\mathbf{\Theta}[0]$
for $k = 1, \dots, K$ ($K$ is the number of samples) do
    draw the sample candidate of parameters $\mathbf{\Theta}^* \sim q(\mathbf{\Theta}^* \mid \mathbf{\Theta}[k-1])$
    draw the initial particles $\hat{\boldsymbol{z}}_1^{(i)} \sim p(\boldsymbol{z}_1)$ for $i = 1, \dots, M$ ($i$ is the particle number of the particle that is the source of resampling)
    calculate the weights of particles $\left\{w_1^{(1)}, w_1^{(2)}, \dots, w_1^{(M)}\right\}$ with Eq. (5.5)
    normalize the weigths of particles $\left\{W_1^{(1)}, W_1^{(2)}, \dots, W_1^{(M)}\right\}$ with Eq. (5.4)
    resample the particles $\left\{\hat{\boldsymbol{z}}_1^{(1)}, \hat{\boldsymbol{z}}_1^{(2)}, \dots, \hat{\boldsymbol{z}}_1^{(M)}\right\}$ according to the normalized weights $\left\{W_1^{(1)}, W_1^{(2)}, \dots, W_1^{(M)}\right\}$
    for $n = 2, \dots, N$ do
        draw the particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \dots, \hat{\boldsymbol{z}}_n^{(M)}\right\}$ at time step $n$ with Eq. (5.3)
        calculate the weights of particles $\left\{w_n^{(1)}, w_n^{(2)}, \dots, w_n^{(M)}\right\}$ with Eq. (5.5)
        normalize the weigths of particles $\left\{W_n^{(1)}, W_n^{(2)}, \dots, W_n^{(M)}\right\}$ with Eq. (5.4)
        resample the particles $\left\{\hat{\boldsymbol{z}}_n^{(1)}, \hat{\boldsymbol{z}}_n^{(2)}, \dots, \hat{\boldsymbol{z}}_n^{(M)}\right\}$ according to the normalized weights $\left\{W_n^{(1)}, W_n^{(2)}, \dots, W_n^{(M)}\right\}$
    end for
    calculate the marginal likelihood $p(\boldsymbol{y}_{1:N} \mid \mathbf{\Theta}^*)$ with Eq. (5.6)
    calculate the acceptance probability $p_{\text{accept}}$ with Eq. (5.1)
    draw a uniform random number $\alpha \sim \mathcal{U}(0, 1)$ ($\mathcal{U}(a, b)$ is a uniform distribution with range $[a, b)$)
    if $\alpha \leq p_{\text{accept}}$ then
        set the sample of parameters $\mathbf{\Theta}[k] \leftarrow \mathbf{\Theta}^*$
    else
        set the sample of parameters $\mathbf{\Theta}[k] \leftarrow \mathbf{\Theta}[k-1]$
    end if
end for
```

determined whether to accept or reject the proposed sample candidate $\mathbf{\Theta}^*$. We show the flow of the PMMH method described above in Algorithm 7.

5.2.2 Proposed method: replica exchange particle marginal Metropolis-Hastings method

In this chapter, we propose the REPMMH method which combines the PMMH method with the replica exchange method [49–51] to improve the problem of initial value dependence in the PMMH method. By employing the REPMMH method, we estimate the marginal posterior distribution of parameters from the time-series observations.

Brief summary of proposed method

We show the schematic diagram of the REPMMH method in Fig. 5.1. In the proposed REPMMH method, we introduce multiple different replicas of parameters $\{\boldsymbol{\Theta}\} = \left\{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \dots, \boldsymbol{\Theta}^{(r)}, \dots, \boldsymbol{\Theta}^{(R)}\right\}$ at temperatures $\boldsymbol{T} = \left[T^{(1)}, T^{(2)}, \dots, T^{(r)}, \dots, T^{(R)}\right]$ into the PMMH method. As shown in the middle part of Fig. 5.1, we employ the PMMH method in parallel at each temperature. In the PMMH method at each temperature $T^{(r)}$, we obtain the respective marginal likelihood $p\left(\boldsymbol{y}_{1:N} \;\middle|\; \boldsymbol{\Theta}^{(r)*}\right)^{\frac{1}{T^{(r)}}}$ by employing the SMC method [Fig. 5.1 (c)] with the respective sample candidate $\boldsymbol{\Theta}^{(r)*}$ proposed in the MH algorithm [Fig. 5.1 (b)].

For each temperature $T^{(r)}$, the SMC method and the MH algorithm are conducted as follows. In the SMC method [Fig. 5.1 (c)], the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \;\middle|\; \boldsymbol{\Theta}^{(r)*}\right)^{\frac{1}{T^{(r)}}}$ is obtained by iterative procedures of predictions, likelihood calculations, and resampling; the latent variables $\boldsymbol{z}_n$ of the current time step n are predicted and the likelihood is calculated for each particle, and resampling is performed according to the calculated likelihoods of particles at each time step. In the MH algorithm [Fig. 5.1 (b)], the sample candidate $\boldsymbol{\Theta}^{(r)*}$ is determined to be accepted or rejected with the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \;\middle|\; \boldsymbol{\Theta}^{(r)*}\right)^{\frac{1}{T^{(r)}}}$. At this time, the target distribution becomes smooth as the temperature becomes high. As a result, it becomes easier to obtain samples from a wide range. Furthermore, exchanges between the samples at different temperatures are conducted in order to realize the

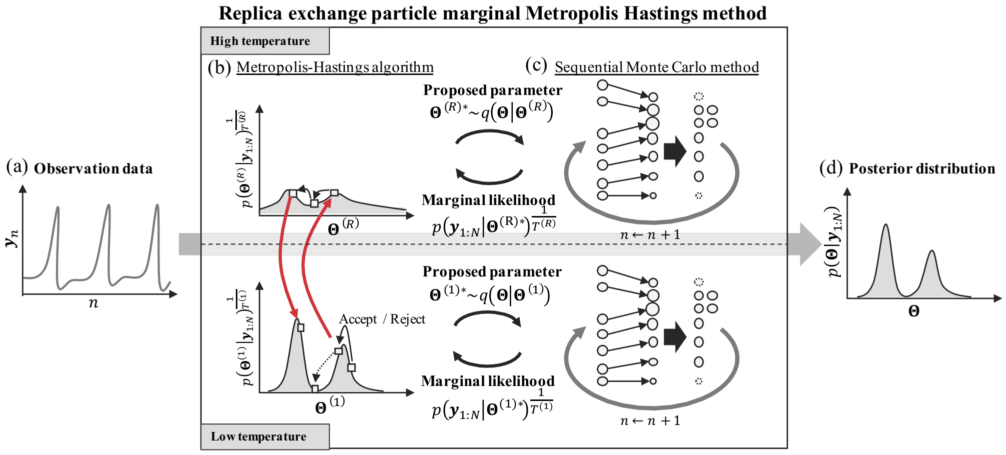

Figure 5.1: Schematic diagrams of the proposed replica exchange particle marginal Metropolis-Hastings (REPMMH) method. (a) The time-series observations $\boldsymbol{y}_{1:N}$ as inputs. (b) and (c) The REPMMH method consisting of (b) the Metropolis-Hastings (MH) algorithms and (c) the sequential Monte Carlo (SMC) methods parallelly conducted at multiple temperatures. In The SMC method, the sample candidate $\boldsymbol{\Theta}^{(r)*}$ proposed by the MH algorithm is used to obtain the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right)^{\frac{1}{T^{(r)}}}$. By the SMC method, the marginalization over time-series of latent variables $\boldsymbol{z}_{1:N}$ is conducted iteratively for time steps $n = 1, 2, \ldots, N$. In the MH algorithm, the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right)^{\frac{1}{T^{(r)}}}$ is used to determine whether to accept or reject the sample candidate. In the REPMMH method, exchanges between samples at different temperatures are considered in order to achieve the transitions that are difficult to achieve with the particle marginal Metropolis-Hastings (PMMH) method. The transitions can be realized by passing through high temperature due to exchange between temperatures as shown by the red arrows in the MH algorithm. (d) The estimated posterior distributions of parameters $\boldsymbol{\Theta}$ as the output.

transitions that are difficult depending on the initial values by the conventional PMMH method.

Introducing the replica exchange method into the PMMH method

Here, we propose the REPMMH method to accurately estimate the distribution of parameters from observed data $\boldsymbol{y}_{1:N}$. In the proposed method, we introduce replicas of parameters $\{\boldsymbol{\Theta}\} = \left\{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(r)}, \ldots, \boldsymbol{\Theta}^{(R)}\right\}$ at different temperatures $\boldsymbol{T} = \left[T^{(1)}, T^{(2)}, \ldots, T^{(r)}, \ldots, T^{(R)}\right]$ and consider the extended joint marginal posterior distribution as follows:

$$\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right) = \prod_{r=1}^{R} \pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right), \tag{5.7}$$

where $\pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ expresses the marginal posterior distribution at temperature $T^{(r)}$, which is expressed by using the original marginal posterior distribution $p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ of the parameter $\boldsymbol{\Theta}^{(r)}$ at the temperature $T^{(1)} = 1.0$ as follows:

$$\pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right) = \frac{1}{z\left(T^{(r)}\right)} p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)^{\frac{1}{T^{(r)}}} \ (r = 1, 2, \ldots, R), \tag{5.8}$$

where $z\left(T^{(r)}\right)$ is a partition function. Note that as expressed in Eq. (5.8), at sufficiently high temperatures, the distribution of parameters becomes closer to a uniform distribution, independent of the values of $\boldsymbol{y}_{1:N}$. The distribution with $T^{(1)} = 1.0$ corresponds to the original marginal posterior distribution $p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$ to be investigated.

The marginal posterior distribution at each temperature $p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ is obtained using Bayes' theorem as follows:

$$p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right) = \frac{p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right) p\left(\boldsymbol{\Theta}^{(r)}\right)}{p\left(\boldsymbol{y}_{1:N}\right)} \ (r = 1, 2, \ldots, R). \tag{5.9}$$

Namely, the marginal posterior distribution $p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ is proportional to a product of a marginal likelihood $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right)$ and a prior distribution $p\left(\boldsymbol{\Theta}^{(r)}\right)$ of parameters $\boldsymbol{\Theta}^{(r)}$. To obtain the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right)$, marginalization of the joint distribution at each temperature should be conducted as follows:

$$p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right) = \int p\left(\boldsymbol{y}_{1:N}, \boldsymbol{z}_{1:N}^{(r)} \,\middle|\, \boldsymbol{\Theta}^{(r)}\right) d\boldsymbol{z}_{1:N}^{(r)} \ (r = 1, 2, \ldots, R), \tag{5.10}$$

where $\boldsymbol{z}_{1:N}^{(r)}$ are the latent variables at the temperature $T^{(r)}$. By performing the SMC method for all the time step at each temperature, the marginalization is conducted numerically.

As shown in Fig. 5.1 (b) and (c), in the proposed method, the SMC method and the MH algorithm are conducted for each temperature. In the SMC method, the marginal likelihood of parameters $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right)$ is determined by the numerical marginalization using candidate of parameters $\boldsymbol{\Theta}^{(r)*}$ proposed in the MH algorithms.

In the MH algorithm, the candidate of parameters $\boldsymbol{\Theta}^{(r)*}$ is determined to be accepted or rejected at each temperature $T^{(r)}$ with the marginal posterior $\pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)*} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ [Fig. 5.1 (b)]. Here, the acceptance probability $p_{\mathrm{accept}}^{(r)}$ at each temperature is calculated as follows:

$$p_{\mathrm{accept}}^{(r)} = \min\left(1, \frac{p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right) p\left(\boldsymbol{\Theta}^{(r)*}\right)}{p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)}[k-1]\right) p\left(\boldsymbol{\Theta}^{(r)}[k-1]\right)} \frac{q\left(\boldsymbol{\Theta}^{(r)}[k-1] \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right)}{q\left(\boldsymbol{\Theta}^{(r)*} \,\middle|\, \boldsymbol{\Theta}^{(r)}[k-1]\right)}\right). \tag{5.11}$$

Moreover, we exchange samples between different temperatures $T^{(r)}$ and $T^{(r+1)}$ according to the exchange probability as follows:

$$p_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*\right) = \min\left(1, R_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*\right)\right), \tag{5.12}$$

$$R_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*\right) = \frac{\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right)}{\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right)}, \tag{5.13}$$

where $\{\boldsymbol{\Theta}\}^*$ is expressed as follows:

$$\{\boldsymbol{\Theta}\}^* = \left\{\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(r+1)}, \boldsymbol{\Theta}^{(r)}, \ldots, \boldsymbol{\Theta}^{(R)}\right\}. \tag{5.14}$$

Note that the exchange probability $p_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*\right)$ corresponds to the Metropolis criterion for proposing to exchange the samples between different temperatures $T^{(r)}$ and $T^{(r+1)}$. By deciding whether to accept or reject the proposed samples $\{\boldsymbol{\Theta}\}^*$ with the Metropolis criterion of Equation (5.12), the transition probability

$W\left(\{\boldsymbol{\Theta}\} \to \{\boldsymbol{\Theta}\}^*\right)$ for the exchange process satisfies the detailed balance condition as follows:

$$
\begin{aligned}
&\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right) W\left(\{\boldsymbol{\Theta}\} \to \{\boldsymbol{\Theta}\}^*\right) \\
&= \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right) p_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*\right) \\
&= \min\left(\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right), \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right)\right) \\
&= \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right) \min\left(\frac{\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right)}{\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right)}, 1\right) \\
&= \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right) p_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^*, \{\boldsymbol{\Theta}\}\right) \\
&= \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right) q\left(\{\boldsymbol{\Theta}\} \,|\, \{\boldsymbol{\Theta}\}^*\right) p_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^*, \{\boldsymbol{\Theta}\}\right) \\
&= \pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\}^* \mid \boldsymbol{y}_{1:N}\right) W\left(\{\boldsymbol{\Theta}\}^* \to \{\boldsymbol{\Theta}\}\right),
\end{aligned}
$$

where $q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right)$ is the proposed probability for $\{\boldsymbol{\Theta}\}^*$ and the proposed probability of the exchange process is symmetric $q\left(\{\boldsymbol{\Theta}\}^* \,|\, \{\boldsymbol{\Theta}\}\right) = q\left(\{\boldsymbol{\Theta}\} \,|\, \{\boldsymbol{\Theta}\}^*\right)$. Thus, since the exchange process in the REPMMH method satisfies the detailed balance condition, the REPMMH method can sample from the distribution $\pi_{\mathrm{EX}}\left(\{\boldsymbol{\Theta}\} \mid \boldsymbol{y}_{1:N}\right)$.

By this exchange process, the REPMMH method makes it possible to improve the problem of initial value dependence in the PMMH method. The sampled distributions of the replica $\pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)$ at higher temperatures become closer to a uniform distribution ideally as follows:

$$
\begin{aligned}
&\lim_{T^{(r)} \to \infty} \pi_{T^{(r)}}\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right) \\
&= \lim_{T^{(r)} \to \infty} \frac{1}{z\left(T^{(r)}\right)} p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)^{\frac{1}{T^{(r)}}} \propto p\left(\boldsymbol{\Theta}^{(r)} \,\middle|\, \boldsymbol{y}_{1:N}\right)^{0} = \text{const.}
\end{aligned}
$$

Therefore, in practice, it becomes possible to escape from local optima at sufficiently high temperatures (Fig. 5.1 (b)). Moreover, the samples may not stay in one local optimum since each replica is exchanged between high temperature and low temperature repeatedly, and we can sample the parameters efficiently. We show the flow of the REPMMH method described above in Algorithm 8.

Algorithm 8 Replica exchange particle marginal Metropolis-Hastings (REPMMH) method

```
initialize the parameters ${\boldsymbol{\Theta}}[0]$
for $k = 1, \ldots, K$ do
    for $r = 1, \ldots, R$ do
        draw the sample candidate of parameters $\boldsymbol{\Theta}^{(r)*} \sim q\left(\boldsymbol{\Theta}^{(r)*} \mid \boldsymbol{\Theta}^{(r)}[k-1]\right)$
        calculate the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \,\middle|\, \boldsymbol{\Theta}^{(r)*}\right)$ by using the SMC method according to Eq. (5.10)
        calculate the acceptance probability $p_{\mathrm{accept}}^{(r)}$ with Eq. (5.11)
        draw a uniform random number $\alpha \sim \mathcal{U}(0,1)$ ($\mathcal{U}(a,b)$ is a uniform distribution with range $[a,b)$)
        if $\alpha \leq p_{\mathrm{accept}}^{(r)}$ then
            set the sample of parameters $\boldsymbol{\Theta}^{(r)}[k] \leftarrow \boldsymbol{\Theta}^{(r)*}$
        else
            set the sample of parameters $\boldsymbol{\Theta}^{(r)}[k] \leftarrow \boldsymbol{\Theta}^{(r)}[k-1]$
        end if
    end for
    choose the replica number $r_{\mathrm{EX}} \leftarrow 1$ or $r_{\mathrm{EX}} \leftarrow 2$ for replica exchange
    repeat
        calculate exchange probability $p_{\mathrm{EX}}(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*)$ with Eq. (5.12) for replica numbers $r_{\mathrm{EX}}$ and $r_{\mathrm{EX}}+1$
        draw a uniform random number $\alpha_{\mathrm{EX}} \sim \mathcal{U}(0,1)$
        if $\alpha_{\mathrm{EX}} \leq p_{\mathrm{EX}}(\{\boldsymbol{\Theta}\}, \{\boldsymbol{\Theta}\}^*)$ then
            exchange replicas $\left(\boldsymbol{\Theta}^{(r_{\mathrm{EX}})}[k], \boldsymbol{\Theta}^{(r_{\mathrm{EX}}+1)}[k]\right) \leftarrow \left(\boldsymbol{\Theta}^{(r_{\mathrm{EX}}+1)}[k], \boldsymbol{\Theta}^{(r_{\mathrm{EX}})}[k]\right)$
        end if
        set the replica number $r_{\mathrm{EX}} \leftarrow r_{\mathrm{EX}} + 2$ for the exchange
    until $r_{\mathrm{EX}} \leq R - 1$
end for
```

Relations among particle Markov chain Monte Carlo methods

We briefly summarize the differences among the conventional particle Markov chain Monte Carlo (PMCMC) methods and proposed REPMMH method that can estimate parameters of a state space model in Table 5.1. The particle-Gibbs (PG) method is another PMCMC method, and it samples latent variables and parameters in a state space model alternately by using Gibbs sampling [8, 32–37]. The PMMH method combines the SMC method with the MH algorithm, whereas the PG method combines the SMC method with Gibbs sampling. While the SMC method is employed to calculate the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ of parameters $\boldsymbol{\Theta}$ in the PMMH method, the SMC method is employed to obtain samples of latent variables $\boldsymbol{z}_{1:N}$ in the PG method [11]. The PMMH method directly targets the marginal posterior distribution $p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$, whereas the PG method targets the joint posterior distribution $p\left(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$ [11]. Note that the SMC method used in the PG method is called the conditional SMC method and uses the previous sample of latent variables $\boldsymbol{z}_{1:N}\left[k-1\right]$ as a particle in the SMC method [11]. Furthermore, advanced versions of the PG method have been proposed, such as the particle-Gibbs with ancestor sampling (PGAS) method [16,18,19] for improving sampling efficiency and the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method proposed in Chapter 3 for improving the initial value dependence. Samples obtained by employing the PMMH method also have a problem of initial value dependence, similar to those obtained by employing the PG method, and it is considered that combining the PMMH method with the replica exchange method would be effective.

5.3 Experiments

In this section, we show that by employing the proposed replica exchange particle marginal Metropolis-Hastings (REPMMH) method for the Izhikevich neuron model [60,61] and Lévy-driven stochastic volatility model [11,63–65], the marginal posterior distribution of parameters $p\left(\boldsymbol{\Theta} \mid y_{1:N}\right)$ can be estimated from observations $y_{1:N}$, and verify whether the REPMMH method can overcome the problem of initial value dependence in the particle marginal Metropolis-Hastings (PMMH) method. Moreover, we compare the sampling efficiency of the REPMMH method with those of the conventional methods, the PMMH method and the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method.

Table 5.1: The PMCMC methods for estimating parameters in a state space model.

Method	**Target**	**Overview**
PG	$p\left(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$	Sample parameters $\boldsymbol{\Theta}$ and latent variables $\boldsymbol{z}_{1:N}$ alternately with Gibbs sampling for targeting the joint posterior distribution $p\left(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$. Note that the SMC method is used for sampling latent variables $\boldsymbol{z}_{1:N}$. The SMC method used in the PG method is called the conditional SMC method and uses the previous sample of latent variables $\boldsymbol{z}_{1:N}[k-1]$ as a particle in the SMC method [11].
PGAS	$p\left(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$	Sample latent variables $\boldsymbol{z}_{1:N}$ not only in the forward direction but also in the backward direction in the PG method [16, 18, 19].
REPGAS	$p\left(\boldsymbol{z}_{1:N}, \boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$	Improve the problem of initial value dependence in the PGAS method by combining the replica exchange method and the PGAS method.
PMMH	$p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$	Sample parameters $\boldsymbol{\Theta}$ with the MH algorithm for targeting directly the marginal posterior distribution $p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$ obtained by marginalization over the distribution of latent variables $\boldsymbol{z}_{1:N}$. Note that the SMC method is used to calculate the marginal likelihood $p\left(\boldsymbol{y}_{1:N} \mid \boldsymbol{\Theta}\right)$ [11].
REPMMH	$p\left(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N}\right)$	Improve the problem of initial value dependence in the PMMH method by combining the replica exchange method and the PMMH method.

5.3.1 Izhikevich neuron model

To verify the effectiveness of the proposed method, we use the Izhikevich neuron model. The Izhikevich neuron model is a computational neuronal model represented by Eqs. (4.1)–(4.4). The system model of the Izhikevich neuron model is represented by Eqs. (4.5)–(4.9) as in Chapter 4. Here, we assume that the observations $y_{1:N}$ are the membrane potentials with Gaussian observation noise, and we estimate the parameters $\boldsymbol{\Theta} = \{a, b, c, d\}$ from only the observations $y_{1:N}$. We use the true parameters $\boldsymbol{\Theta} = \{a, b, c, d\} = \{0.02, 0.2, -65, 6\}$ and the number of data $N = 5.0 \times 10^2$ to generate data. In the system model, the means and the variances of the Gaussian noise are $\left\{\mu_v, \sigma_v^2\right\} = \{0, 0.25\}$ and $\left\{\mu_u, \sigma_u^2\right\} = \left\{0, 10^{-4}\right\}$. In the observation model, the mean and the variance of the Gaussian noise are $\left\{\mu_y, \sigma_y^2\right\} = \{0, 1\}$. We show the generated data from the Izhikevich neuron model in Fig. 5.2. In Fig. 5.2, complex spike activities with different inter-spike intervals and different peaks are seen in response to external inputs. We assume that only one dimensional time series of observed data y_n and external inputs can be used for estimating underlying parameters while the latent dynamics are governed by two dimensional nonlinear dynamical systems with four parameters $\boldsymbol{\Theta} = \{a, b, c, d\}$. We employ the REPMMH method and the conventional methods, the PMMH method and the REPGAS method, to the generated data in Fig. 5.2 to estimate the posterior distribution of the parameters $\boldsymbol{\Theta} = \{a, b, c, d\}$. In all methods, the initial values of the parameters $\boldsymbol{\Theta}\,[0]$ are $\{a, b, c, d\} = \{0.025, 0.15, -60, 5.5\}$, the number of samples K is 10^6, the number of burn-in samples $K_{\text{burn-in}}$ is 10^6, and the number of particles M is 50. In the REPMMH method and the REPGAS method, the number of temperatures R is 64.

Figure 5.3 shows the estimated posterior distribution of parameters $p\left(\boldsymbol{\Theta} \mid y_{1:N}\right)$ obtained by employing the PMMH method. In each graph the vertical axis expresses the value of the probability density function, while the horizontal axis expresses the values of parameters a, b, c, and d. Furthermore, the solid lines represent the true values, the dashed lines represent the initial values, and the histograms represent the estimated posterior distributions of the parameters. From Fig. 5.3, we find that a peak of the estimated posterior distribution of parameter d, $p\left(d \mid y_{1:N}\right)$, is located around its true value $d = 6.0$. However the maximum values of the estimated posterior distribution of the other three parameters, a, b, and c remain around their initial values ($a = 0.025$, $b = 0.15$, $c = -60$), which are far from their true values ($a = 0.020$, $b = 0.20$, $c = -65$). Thus the joint posterior distribution of four

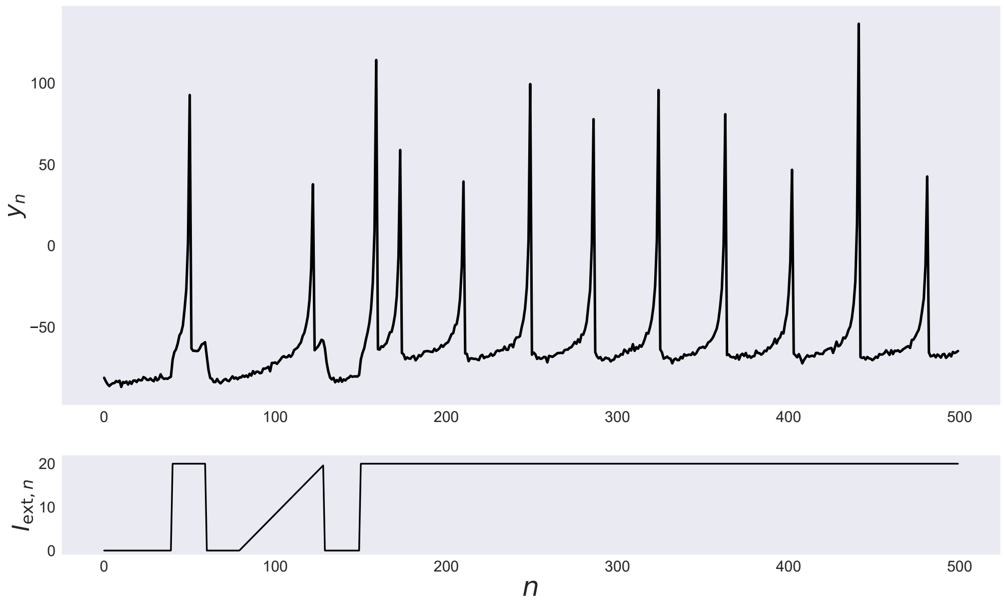

Figure 5.2: Observations and external inputs of Izhikevich neuron model used to evaluate the proposed REPMMH method. Observed membrane potential of the Izhikevich neuron model y_n (top) in response to input current $I_{\mathrm{ext},n}$ (bottom) are shown.

parameters are found to be not adequately estimated. From this result, the samples in the PMMH method is consider to remain in the local optimum since the initial values are far from the true value.

Figure 5.4 shows the estimated posterior distribution of parameters $p\left(\boldsymbol{\Theta} \mid y_{1:N}\right)$ obtained by employing the REPMMH method. From Fig. 5.4, we find that the maximum values of estimated posterior distribution of parameters are located around true values ($a = 0.020$, $b = 0.20$, $c = -65$, $d = 6.0$) even though the initial values of parameters ($a = 0.025$, $b = 0.15$, $c = -60$, $d = 5.5$) are set to be far from the true values. This improvement of estimation accuracy would be induced by combining the replica exchange method. In the proposed REPMMH method, it is easier to obtain samples from a wider range since the replica exchange method allows samples to pass through high temperatures. From these results, we find that the problem of initial value dependence in the PMMH method is improved by employing the

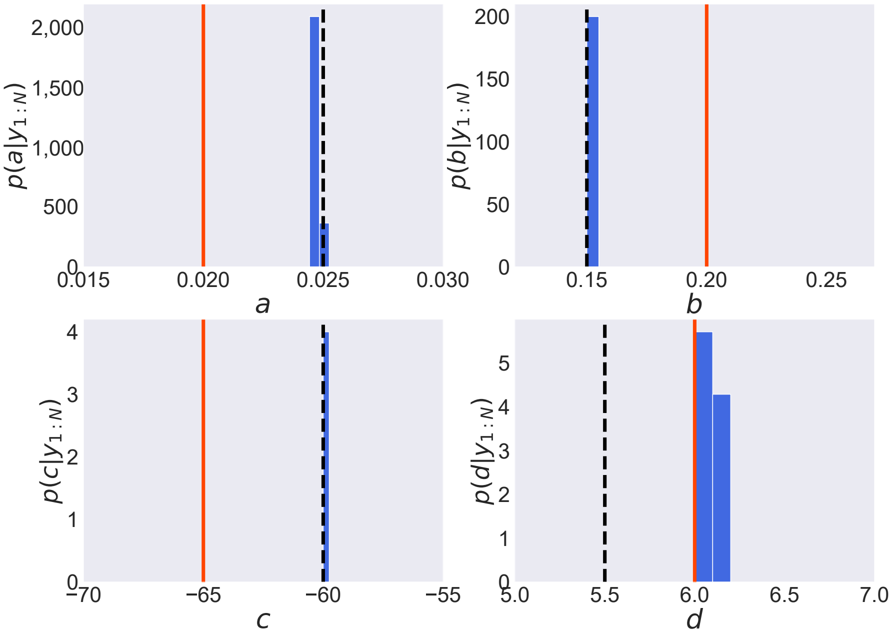

Figure 5.3: Estimated posterior distributions obtained by employing the PMMH method in the Izhikevich neuron model. In each graph, the estimated probability density function of parameter (a, b, c, and, d) is shown by the blue histogram. The red solid and black dashed lines express the true and initial values, respectively.

proposed method.

Moreover, Fig. 5.5 shows the estimated posterior distribution of parameters $p\left(\boldsymbol{\Theta} \mid y_{1:N}\right)$ obtained by employing the REPGAS method. As shown in this figure, the distributions are estimated almost the same as those obtained by employing the REPMMH method, which indicates that the true values are estimated properly.

In order to investigate efficiency of sampling parameters in the proposed method and existing methods, we show in Fig. 5.6 the autocorrelation function results calculated using the samples of the PMMH method, the REPGAS method, and the proposed REPMMH method. In all the parameters a, b, c, and d, the decay of the autocorrelation in the REPMMH samples is faster using the REPMMH sam-

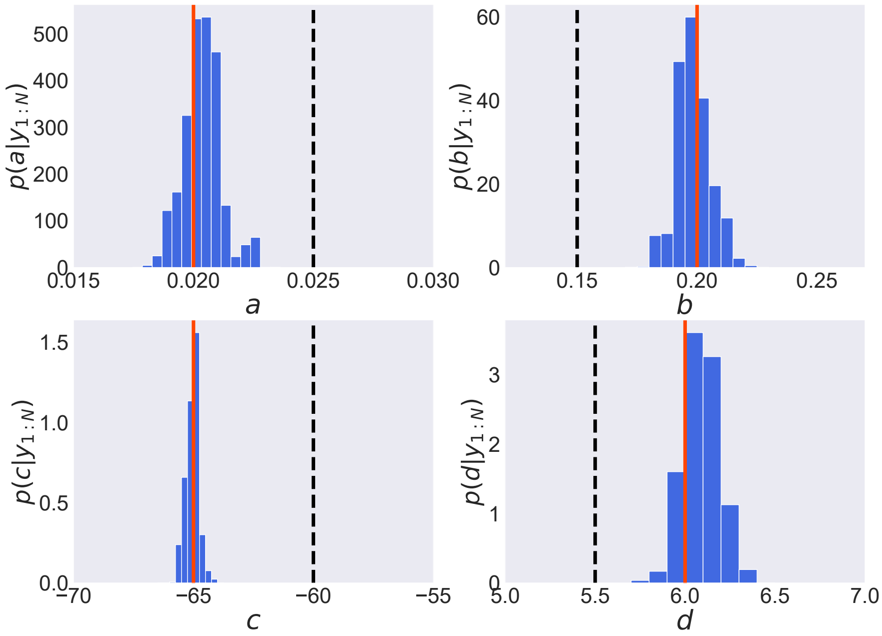

Figure 5.4: Estimated posterior distributions obtained by employing the REPMMH method in the Izhikevich neuron model. See also the captions of the figure and subfigures for Fig 5.3.

ples than those calculated using the PMMH samples and the REPGAS samples. In particular, the introduction of the replica exchange method drastically improves the sampling efficiency of the PMMH method. Even if we compare two PMCMC methods with the replica exchange method, the proposed method provides more efficient sampling. Thus, we find that the sampling efficiency of the proposed REPMMH method is higher than that of the PMMH method and the REPGAS method. The time constant of the autocorrelation function has a strong influence on the convergence time of the PMCMC method. The time constants of the autocorrelation functions for the REPMMH samples are around 20 for all parameters a, b, c and d, while those of the autocorrelation functions for the PMMH samples are more than

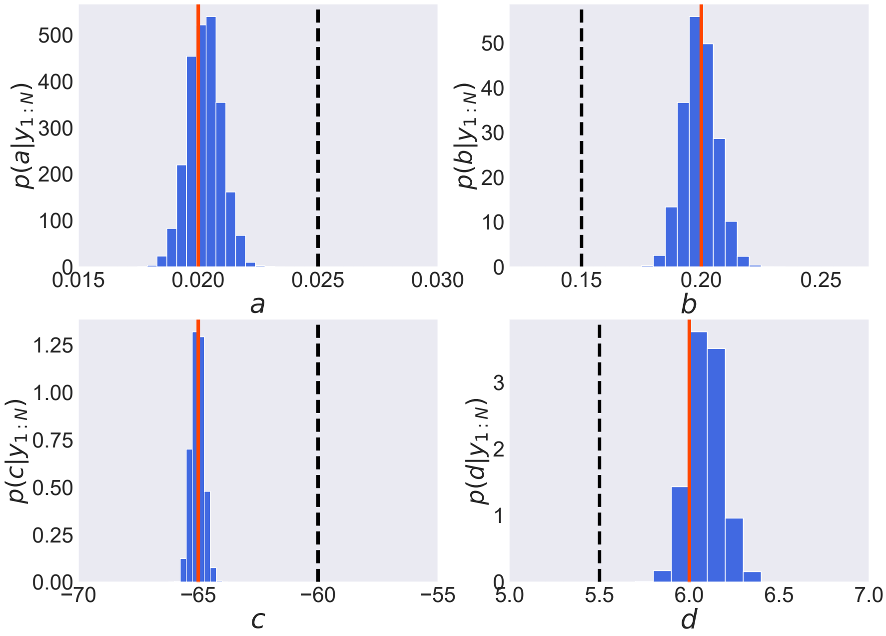

Figure 5.5: Estimated posterior distributions obtained by employing the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method in the Izhikevich neuron model. See also the captions of the figure and subfigures for Fig 5.3.

10^5, as shown in Figure 5.6. Since the computational cost of the exchange process in the REPMMH method is very small compared to the computational cost of the SMC method, the computational cost of the REPMMH method is approximately $R = 64$ times the computational cost of the PMMH method. Nevertheless, the REPMMH method drastically improves the sampling efficiency compared to the increase in the computational cost; the REPMMH method is $R = 64$ times more computationally expensive than the PMMH method, while the effective sample size of the REPMMH method is much larger (around 10^3 times larger) than that of the PMMH method.

When the same number of temperatures and particles is used, the REPGAS method is more computationally expensive than the REPMMH method since the

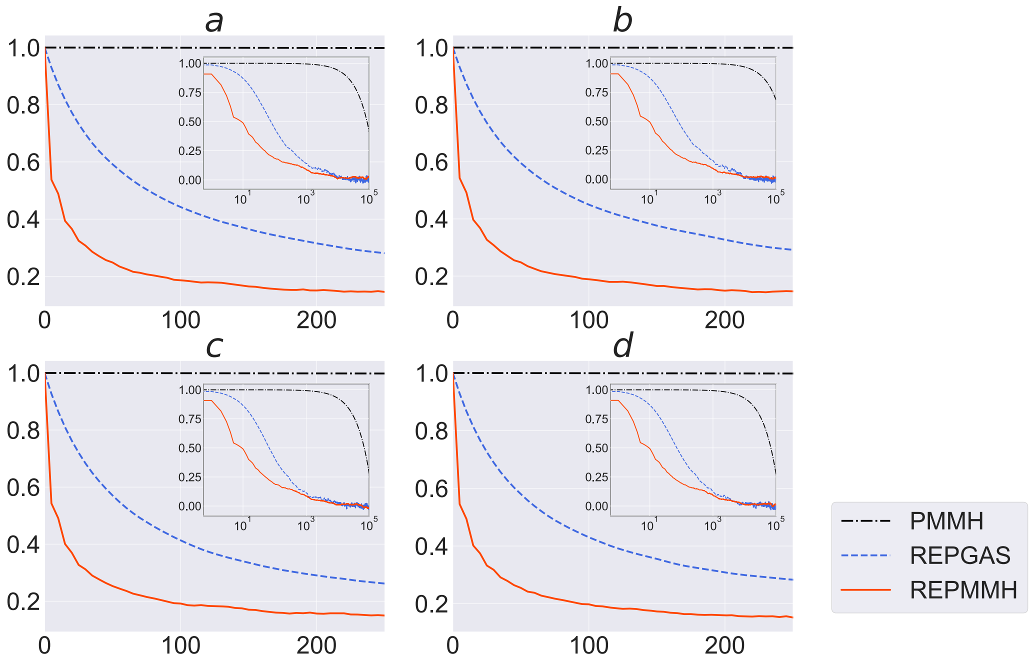

Figure 5.6: Autocorrelation as a function of the lag length for parameters a, b, c, and d in the Izhikevich neuron model. Results for the PMMH method (black dashed-dotted line), the REPGAS method (blue dashed line), and the REPMMH method (red solid line) are shown. Each inset figure represents the result when the horizontal axis is the logarithmic scale. In results obtained by the REPGAS method and the REPMMH method, samples at $T^{(1)} = 1.0$ are used.

REPGAS method requires the sampling of the latent variables $\boldsymbol{z}_{1:N}$ and the ancestor sampling, which considers sampling of the latent variables $\boldsymbol{z}_{1:N}$ not only in the forward direction but also in the backward direction in the conditional SMC method. Nevertheless, the REPMMH method has high sampling efficiency compared to the REPGAS method. Thus, we find that the sampling efficiency of the REPMMH method is higher than that of the conventional methods.

Moreover, in order to evaluate the influence of the number of temperatures R and the number of particles M on the estimated results, we compare the estimated results in various settings. We show the estimated results with the numbers of temperatures $R = 1$, 4, 16 and 64 in Table 5.2. Table 5.2 shows the mode values of the estimated distributions, the standard deviations (Std) of the estimated distributions and the values of autocorrelation functions (ACF) with the lag length 30 for the numbers of temperatures $R = 1$, 4, 16 and 64. Note that the numbers of particles M are 50 in all cases and the maximum value of temperature is fixed at $T^{(R)} = 1.1^{63}$ for $R > 1$.

As mentioned above, for the number of temperatures $R = 1$, we can estimate the parameter d around the true value $d = 6.0$, while the other three parameters a, b and c remain at their initial values ($a = 0.025$, $b = 0.15$, $c = -60$). We also find that the samples of parameters a, b and c cannot move enough from their initial values since the values of the standard deviations are very small. For the number of temperatures $R = 4$, we find the samples can escape the local optima and we can estimate the true values of all parameters a, b, c and d accurately due to the high temperatures that allow escape from the local optima. However, since the values of the autocorrelation functions are close to 1.0, we need a large number of samples in order to estimate the shape of the distribution $p(\boldsymbol{\Theta} \mid y_{1:N})$. On the other hand, we find that the values of the autocorrelation functions are smaller for $R = 16$ and 64.

Table 5.2: The estimated results with the numbers of temperatures $R = 1$, 4, 16 and 64.

Parameter		$R = 1$	$R = 4$	$R = 16$	$R = 64$
$a = 0.020$	Mode	0.0251	0.0200	0.0205	0.0205
	Std	5.5×10^{-5}	6.9×10^{-4}	6.7×10^{-4}	7.8×10^{-4}
	ACF	0.9999	0.9914	0.5175	0.3074
$b = 0.20$	Mode	0.155	0.200	0.200	0.200
	Std	3.0×10^{-4}	7.2×10^{-3}	7.3×10^{-3}	7.0×10^{-3}
	ACF	0.9999	0.9919	0.5773	0.3082
$c = -65$	Mode	-60.0	-64.75	-65.0	-65.0
	Std	2.1×10^{-3}	2.8×10^{-1}	2.5×10^{-1}	2.6×10^{-1}
	ACF	0.9999	0.9926	0.5359	0.3117
$d = 6.0$	Mode	6.10	6.10	6.05	6.05
	Std	2.0×10^{-2}	8.9×10^{-2}	9.8×10^{-2}	9.8×10^{-2}
	ACF	0.9999	0.9928	0.5222	0.3176

We show the estimated results with the numbers of particles $M = 10$, 20, 30, 40 and 50 in Table 5.3. Note that the numbers of temperatures R are 64 in all cases. For the numbers of particles $M = 10$ and 20, the estimated values of the parameters a, b, c and d are far from the true values ($a = 0.020$, $b = 0.20$, $c = -65$, $d = 6.0$). We consider that these results are due to the low approximation accuracy of the marginal likelihood $p\left(y_{1:N} \mid \mathbf{\Theta}\right)$ in the SMC method with too small numbers of particles. For the numbers of particles $M \geq 30$, we can estimate the true values of parameters a, b, c and d. Since there is no significant difference between the mode values, the standard deviations and the values of the autocorrelation functions for the numbers of particles $M = 30$, 40 and 50, we consider that the number of particles M is sufficient for this problem if it is above 30.

Table 5.3: The estimated results with the numbers of particles $M = 10$, 20, 30, 40 and 50.

Parameter		$M = 10$	$M = 20$	$M = 30$	$M = 40$	$M = 50$
$a = 0.020$	Mode	0.0220	0.0210	0.0200	0.0200	0.0205
	Std	5.9×10^{-4}	7.5×10^{-4}	7.3×10^{-4}	6.7×10^{-4}	7.8×10^{-4}
	ACF	0.5548	0.2707	0.3072	0.3391	0.3074
$b = 0.20$	Mode	0.195	0.190	0.200	0.195	0.200
	Std	4.9×10^{-3}	8.3×10^{-3}	7.1×10^{-3}	5.9×10^{-3}	7.0×10^{-3}
	ACF	0.2738	0.2611	0.2932	0.3352	0.3082
$c = -65$	Mode	-60.75	-64.50	-65.00	-65.00	-65.00
	Std	2.9×10^{-1}	3.7×10^{-1}	2.7×10^{-1}	2.8×10^{-1}	2.6×10^{-1}
	ACF	0.9223	0.3071	0.2793	0.3701	0.3117
$d = 6.0$	Mode	6.50	6.10	6.10	6.05	6.05
	Std	1.4×10^{-1}	8.2×10^{-2}	1.0×10^{-1}	9.3×10^{-2}	9.8×10^{-2}
	ACF	0.6096	0.2750	0.3220	0.2985	0.3176

5.3.2 Lévy-driven stochastic volatility model

Next, we also verify the effectiveness of the proposed method using the Lévy-driven stochastic volatility model [11, 63–65]. In this model, the dynamics of logarithm of asset price $y^*(t)$ is represented by the following differential equation:

$$dy^*(t) = \left\{\mu + \beta\sigma^2(t)\right\} dt + \sigma(t)\, dB(t), \tag{5.15}$$

where μ is the drift parameter and β is the risk premium. $B(t)$ is the Brownian motion and $\sigma^2(t)$ represents the volatility. The dynamics of the volatility $\sigma^2(t)$ is modeled by the following Lévy-driven Ornstein-Unlenbeck process:

$$d\sigma^2(t) = -\lambda\sigma^2(t)\,dt + dz(\lambda t), \tag{5.16}$$

where λ is a positive constant and $z(t)$ is a non-Gaussian Lévy process with positive increments. The observation at the time step n, y_n, in this model is obtained by the following Gaussian distribution:

$$y_n \sim \mathcal{N}\left(\mu\Delta + \beta\sigma_n^2, \sigma_n^2\right), \tag{5.17}$$

where Δ is the length of the time interval.

The stochastic volatility models are numerically investigated by using discretized dynamical models [11, 63, 64], and the estimation algorithm for parameters of the stochastic volatility models have been investigated using such the discretized models [11]. The integrated volatility σ_n^2 at the time step n is calculated as follows:

$$\begin{aligned} \sigma_n^2 &= \int_{(n-1)\Delta}^{n\Delta} \sigma^2(u)\,du \\ &= \lambda^{-1}\left[z(\lambda n\Delta) - \sigma^2(n\Delta) - z\{\lambda(n-1)\Delta\} + \sigma^2\{(n-1)\Delta\}\right], \end{aligned} \tag{5.18}$$

where $\sigma^2(n\Delta)$ and $z(\lambda n\Delta)$ are respectively represented as follows:

$$\sigma^2(n\Delta) = \exp(-\lambda\Delta)\,\sigma^2\{(n-1)\Delta\} + \eta_{\sigma,n}, \tag{5.19}$$

$$z(\lambda n\Delta) = z\{\lambda(n-1)\Delta\} + \eta_{z,n}. \tag{5.20}$$

Here, we address the case where the volatility $\sigma^2(t)$ follows a tempered stable marginal distribution [64]. Following [3, 64], $\eta_{\sigma,n}$ and $\eta_{z,n}$ are obtained as follows:

$$\eta_{\sigma,n} = \sum_{i=1}^{\infty} \min\left(\left(\frac{a_i\kappa}{A\lambda\Delta}\right)^{-1/\kappa}, e_i v_i^{1/\kappa}\right) \exp\left(-\lambda\Delta r_i\right) + \sum_{i=1}^{N(\lambda\Delta)} c_i \exp\left(-\lambda\Delta r_i^*\right), \quad (5.21)$$

$$\eta_{z,n} = \sum_{i=1}^{\infty} \min\left(\left(\frac{a_i\kappa}{A\lambda\Delta}\right)^{-1/\kappa}, e_i v_i^{1/\kappa}\right) + \sum_{i=1}^{N(\lambda\Delta)} c_i, \quad (5.22)$$

where $A = 2^{\kappa}\delta\kappa^2/\Gamma\left(1-\kappa\right)$, $a_1 < a_2 < \ldots$ are arrival times of a Poisson process with intensity 1, $e_1, e_2, \ldots$ are independent and identically distributed exponential random variables with mean $2\gamma^{-1/\kappa}$, and $v_1, v_2, \ldots$, $r_1, r_2, \ldots$, and $r_1^*, r_2^*, \ldots$ are standard uniform random variables. $c_1, c_2, \ldots$ are obtained from a gamma distribution with the shape parameter $1-\kappa$ and the scale parameter $2\gamma^{-1/\kappa}$, and $N\left(\lambda\Delta\right)$ is obtained from a Poisson distribution with mean $\lambda\Delta\delta\gamma\kappa$. Here, κ, δ, γ, and λ are the parameters $\mathbf{\Theta} = \{\kappa, \delta, \gamma, \lambda\}$ to be estimated.

In this section, we employ the proposed method and the PMMH method to the stochastic volatility model. The PMMH-based methods, including the proposed REPMMH method, can be applied to complex models like the Lévy-driven stochastic volatility model, as long as the probability density of the observation model can be calculated. On the other hand, the PG-based method is difficult to apply to the stochastic volatility model since it is necessary to calculate the probability density of the system model in addition to that of the observation model [11]. Following [11], we use the true parameters $\mathbf{\Theta} = \{\kappa, \delta, \gamma, \lambda\} = \{0.5, 1.41, 2.83, 0.1\}$, the number of data $N = 4.0 \times 10^2$, and the time interval of length $\Delta = 1.0$ to generate data. In order to estimate the parameters $\mathbf{\Theta}$, we use the initial values of the parameters $\mathbf{\Theta}[0] = \{0.25, 7.41, 9.83, 1.5\}$, the number of samples $K = 1.5 \times 10^5$, the number of burn-in samples $K_{\mathrm{burn-in}} = 10^5$, and the number of particles $M = 200$. In the REPMMH method, the number of temperatures R is 64 in both the REPMMH method and the PMMH method.

Figure 5.7 shows the estimated posterior distribution of parameters $p\left(\mathbf{\Theta} \mid y_{1:N}\right)$

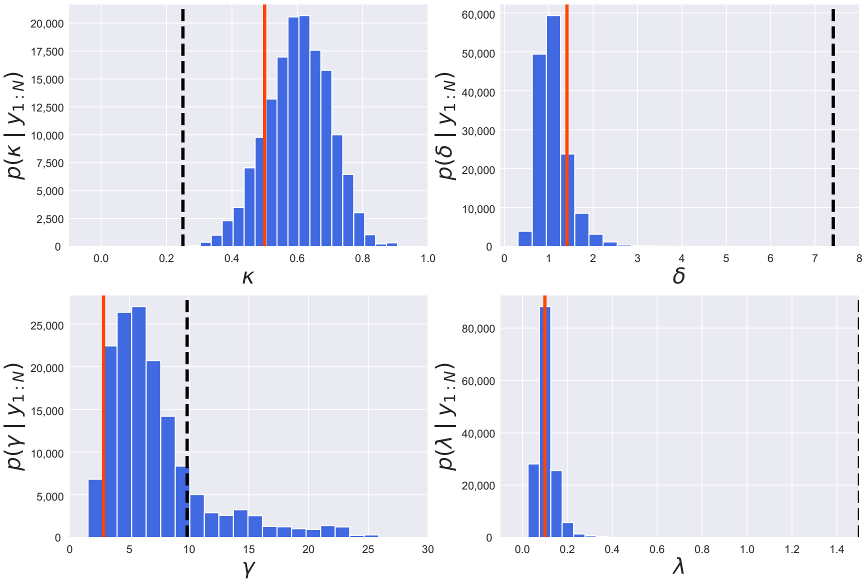

Figure 5.7: Estimated posterior distributions obtained by employing the PMMH method in the Lévy-driven stochastic volatility model. In each graph, the estimated probability density function of parameter (κ, δ, γ, and λ) is shown by the blue histogram. The red solid and black dashed lines express the true and initial values, respectively.

obtained by employing the PMMH method. From Fig. 5.7, we find that a peak of the estimated posterior distribution of parameter λ, $p\left(\lambda \mid y_{1:N}\right)$, is located around its true value $\lambda = 0.1$. However, the maximum values of the estimated posterior distribution of the other three parameters, κ, δ, and γ are far from their true values ($\kappa = 0.5$, $\delta = 1.41$, $\gamma = 2.83$). Thus, the joint posterior distribution of four parameters are found to be not adequately estimated. It is consider that the target distribution is not reached with a small number of samples since the sampling efficiency of the PMMH method is low.

Figure 5.8 shows the estimated posterior distributions of parameters $p\left(\boldsymbol{\Theta} \mid y_{1:N}\right)$ obtained by employing the REPMMH method. From Fig. 5.8, we find that the true values of parameters $\boldsymbol{\Theta} = \{\kappa, \delta, \gamma, \lambda\} = \{0.5, 1.41, 2.83, 0.1\}$ are estimated appropriately by using the same number of samples and the same initial values $\boldsymbol{\Theta}[0] = \{0.25, 7.41, 9.83, 1.5\}$ used in the PMMH method. The results in Figs. 5.7 and 5.8 show that the REPMMH method has higher sampling efficiency than the PMMH method. Moreover, we show in Fig. 5.9 the autocorrelation function results calculated using the samples of the PMMH method and the REPMMH method. In all parameters κ, δ, γ, and λ, the decay of the autocorrelation is faster using the REPMMH samples than that is using the PMMH samples. As shown in Figure 5.9, the time constants of the autocorrelation functions for the REPMMH samples are less than 15 for all parameters κ, δ, γ and λ, while the time constant of the autocorrelation functions for the PMMH samples for the parameter γ is more than 3.0×10^3. As mentioned above, since the computational cost of the REPMMH method is approximately $R = 64$ times the computational cost of the PMMH method, the REPMMH method improves the sampling efficiency compared to the increase in the computational cost.

5.4 Summary

In this chapter, we have proposed the replica exchange particle marginal Metropolis-Hastings (REPMMH) method in order to estimate the marginal posterior distribution of parameters $p(\boldsymbol{\Theta} \mid \boldsymbol{y}_{1:N})$ of the state space model. The proposed method can be applied to complex models like the Lévy-driven stochastic volatility model even if the probability densities of the system models cannot be calculated explicitly. By the proposed method, we introduce the exchange between samples of model parameters $\boldsymbol{\Theta}$ at different temperatures and realize the efficient sampling method for model parameters governing the nonlinear dynamical systems.

Using nonlinear dynamical models such as the Izhikevich neuron model and Lévy-driven stochastic volatility model, we showed that the proposed REPMMH method can improve the problem of initial value dependence of the particle marginal

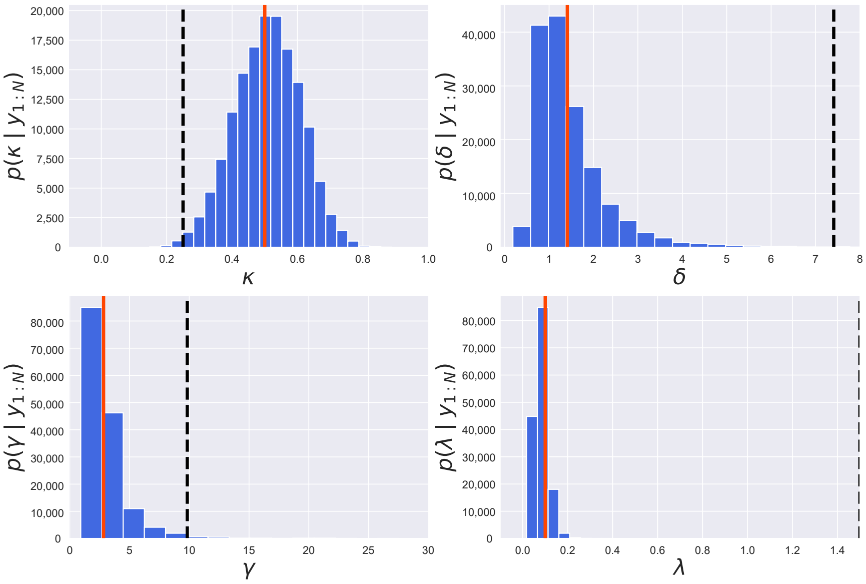

Figure 5.8: Estimated posterior distributions obtained by employing the REPMMH method in the Lévy-driven stochastic volatility model. See also the captions of the figure and subfigures for Fig 5.7.

Metropolis-Hastings (PMMH) method. The results have shown that the proposed REPMMH method accurately estimates the marginal posterior distribution of parameters. Moreover, by comparing the autocorrelation functions of the obtained samples, it has been also shown that the proposed REPMMH method can sample more efficiently than the conventional methods. In the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method, the next sample of latent variables is obtained under the strong influence if the current sample of latent variables. On the other hand, in the REPMMH method, the correlation of the latent variables between current and next steps is low since the REPMMH method only calculates the marginal likelihood of the next step regardless of the latent variables obtained in the current step. Therefore, it is considered that the REPMMH method can sample

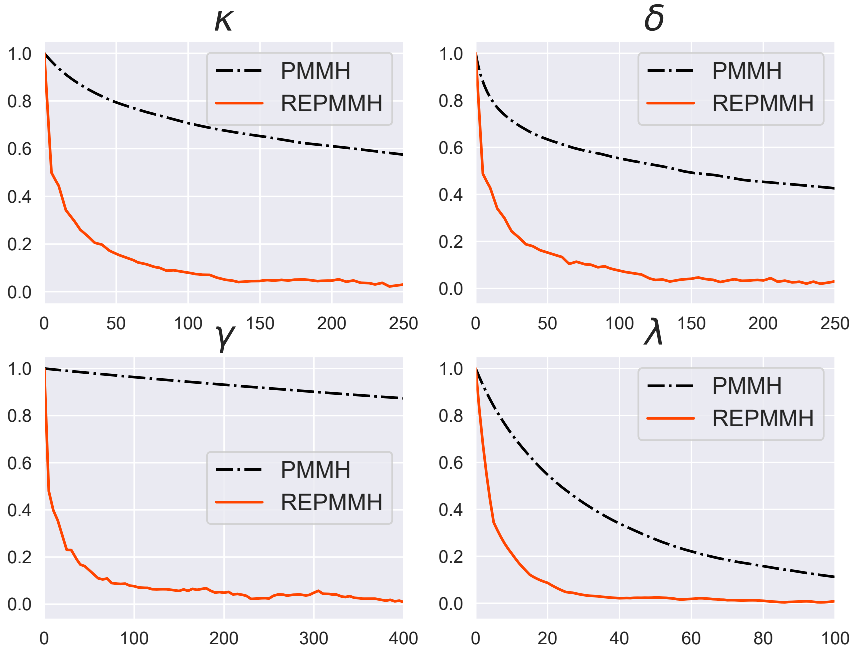

Figure 5.9: Autocorrelation as a function of the lag length for parameters κ, δ, γ, and λ in the Lévy-driven stochastic volatility model. Results for the PMMH method (black dashed-dotted line) and the REPMMH method at $T^{(1)} = 1.0$ (red solid line) are shown.

parameters more efficiently than the REPGAS method.

Chapter 6

Conclusion

In this dissertation, we proposed the new particle Markov chain Monte Carlo (PMCMC) methods to more precisely estimate latent variables and parameters in state space models. First, we proposed the replica exchange particle-Gibbs with ancestor sampling (REPGAS) method to estimate joint posterior distributions of latent variables and parameters of state space models. The REPGAS method combined the particle-Gibbs with ancestor sampling (PGAS) method, which is one of the extended versions of the particle-Gibbs (PG) methods, with the replica exchange method used in the field of statistical physics. The proposed REPGAS method has introduced the extended variables corresponding to temperatures, and conduct MCMC methods in parallel at different temperatures. Moreover, the REPGAS method realized the local precise search and global search simultaneously by exchanging samples between different temperatures. We verified that the proposed REPGAS method can improve the problem of the initial value dependence and the sampling efficiency in the PGAS method by conducting the experiment with the nonlinear benchmark state space model.

Next, we proposed the method to estimate the nonlinear neural dynamics from spike-train data based on the REPGAS method. We formulated the generative process of the spike-train data as the state space model based on the Izhikevich neuron model. Moreover, we formulated the process observing the spike-train data according to the latent membrane potential by using the Poisson distribution. Furthermore, we verified that the proposed method can be used to estimate the latent variables and the parameters that can represent the important responses of neurons through validation experiments using simulated data.

Finally, we proposed the replica exchange particle marginal Metropolis-Hastings (REPMMH) method to estimate the posterior distribution of the parameters marginalized out the distribution of latent variables. The conventional particle marginal Metropolis-Hastings method also has the problem of the initial value dependence as well as the PG method. Therefore, we proposed the REPMMH method by combining the particle marginal Metropolis-Hastings (PMMH) method with the replica exchange method. We verified that the REPMMH method can improve the problem of the initial value dependence in the PMMH method by conducting the experiments with the Izhikevich neuron model and the stochastic volatility model. Moreover, we showed that the REPMMH method has higher sampling efficiency than the REPGAS method and the PMMH method.

In the REPGAS method, the next sample of latent variables is obtained under the strong influence to the current sample of latent variables. On the other hand, in the REPMMH method, the correlation of samples of the latent variables between current and next steps is low since the REPMMH method only calculates the marginal likelihood of the next step regardless of the latent variables obtained in the current step. Therefore, it is considered that the REPMMH method has higher sampling efficiently than the REPGAS method in the estimation of the parameters. However, since the REPMMH method marginalizes out the latent variables, it is considered that the REPGAS method is also an effective method when the joint posterior distribution of the latent variables and parameters need to be estimated.

In this dissertation, we conducted the experiments by using the fix temperature distributions. By optimizing the temperature distributions, it may be possible that the sampling efficiencies of the proposed REPGAS method and REPMMH method are higher than the results in this dissertation. Furthermore, the optimum temperature distributions would be different between the REPGAS method and the REPMMH methods. Therefore, it is important to consider optimization problems for temperature distributions in the proposed REPGAS method and the REPMMH method. Moreover, although we conducted the experiments by using simulated data with no missing values in this dissertation, both the REPGAS method and the REPMMH method can be applied to time-series data with missing values. It is important to evaluate the robustness of the proposed methods to missing values for application to real data. We leave these as future works.

Bibliography

[1] M. L. Andrade Netto, L. Gemenao, and M. J. Mendes, "A New Spline Algorithm for Non-Linear Filtering of Discrete Time Systems," IFAC Proceedings Volumes, 11(1), 2123–2130 (1978).

[2] Z. Ghahramani and G. E. Hinton, "Parameter Estimation for Linear Dynamical Systems," University of Toronto Technical Report CRG-TR-96-2 (1996).

[3] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," Statistics and Computing, 10(3), 197–208 (2000).

[4] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer-Verlag, 2001).

[5] G. Kitagawa, "A Self-Organizing State-Space Model," Journal of the American Statistical Association, 93(443), 1203–1215 (1998).

[6] R. Meyer and N. Christensen, "Fast Bayesian Reconstruction of Chaotic Dynamical Systems via Extended Kalman Filtering," Physical Review E, 65(1), 016206 (2001).

[7] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A State Space Framework for Automatic Forecasting Using Exponential Smoothing Methods," International Journal of Forecasting, 18(3), 439–454 (2002).

[8] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Heidelberg, 2006).

[9] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski, "Spike Inference from Calcium Imaging Using Sequential Monte Carlo Methods," Biophysical Journal, 97(2), 636–655 (2009).

[10] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, "Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging," Journal of Neurophysiology, 104(6), 3691–3704 (2010).

[11] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov Chain Monte Carlo Methods," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(3), 269–342 (2010).

[12] T. Tsunoda, T. Omori, H. Miyakawa, M. Okada, and T. Aonishi, "Estimation of Intracellular Calcium Ion Concentration by Nonlinear State Space Modeling and Expectation-Maximization Algorithm for Parameter Estimation," Journal of the Physical Society of Japan, 79(12), 124801 (2010).

[13] L. Meng, M. A. Kramer, and U. T. Eden, "A Sequential Monte Carlo Approach to Estimate Biophysical Neural Models from Spikes," Journal of neural engineering, 8(6), 065006 (2011).

[14] F. Kwasniok, "Estimation of Noise Parameters in Dynamical System Identification with Kalman Filters," Physical Review E, 86(3), 036214 (2012).

[15] L. Paninski, M. Vidne, B. DePasquale, and D. G. Ferreira, "Inferring Synaptic Inputs Given a Noisy Voltage Trace via Sequential Monte Carlo Methods," Journal of Computational Neuroscience, 33(1), 1–19 (2012).

[16] F. Lindsten, T. Schön, and M. I. Jordan, "Ancestor Sampling for Particle Gibbs," Advances in Neural Information Processing Systems 25, 2600–2608 (2012).

[17] S. Henriksen, A. Wills, T. B. Schön, and B. Ninness, "Parallel Implementation of Particle MCMC Methods on a GPU," IFAC Proceedings Volumes, 45(16), 1143–1148 (2012).

[18] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian Inference and Learning in Gaussian Process State-Space Models with Particle MCMC," Advances in Neural Information Processing Systems 26 (2013).

[19] F. Lindsten, M. I. Jordan, and T. B. Schön, "Particle Gibbs with Ancestor Sampling," Journal of Machine Learning Research, 15(63), 2145–2184 (2014).

[20] T. Omori, T. Kuwatani, A. Okamoto, and K. Hukushima, "Bayesian Inversion Analysis of Nonlinear Dynamics in Surface Heterogeneous Reactions," Physical Review E, 94(3), 033305 (2016).

[21] T. Omori, T. Sekiguchi, and M. Okada, "Belief Propagation for Probabilistic Slow Feature Analysis," Journal of the Physical Society of Japan, 86(8), 084802 (2017).

[22] D. Dreano, P. Tandeo, M. Pulido, B. Ait - El - Fquih, T. Chonavel, and I. Hoteit, "Estimating Model - Error Covariances in Nonlinear State - Space Models Using Kalman Smoothing and the Expectation–Maximization Algorithm," Quarterly Journal of the Royal Meteorological Society, 143(705), 1877–1885 (2017).

[23] R. D. Snyder, J. K. Ord, A. B. Koehler, K. R. McLaren, and A. N. Beaumont, "Forecasting Compositional Time Series: A State Space Approach," International Journal of Forecasting, 33(2), 502–512 (2017).

[24] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, “Deep State Space Models for Time Series Forecasting,” Advances in Neural Information Processing Systems 31, 7785–7794 (2018).

[25] Wang, P, Yang, M, Peng, Y, Zhu, J, Ju, R, and Yin, Q, “Sensor Control in Anti-Submarine Warfare–A Digital Twin and Random Finite Sets based Approach,” Entropy, 21(8), 767 (2019).

[26] Shapovalova, Y, ““Exact” and Approximate Methods for Bayesian Inference: Stochastic Volatility Case Study,” Entropy 23(4), 466 (2021).

[27] P. Gregory, *Bayesian Logical Data Analysis for the Physical Sciences* (Cambridge University Press, 2010).

[28] C. F. J. Wu, “On the Convergence Properties of the EM Algorithm,” The Annals of Statistics, 11(1), 95–103 (1983).

[29] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Its Extensions* (Wiley, New York, 1997).

[30] E. Tzoreff and A. J. Weiss, “Expectation-Maximization Algorithm for Direct Position Determination,” Signal Processing, 133, 32–39 (2017).

[31] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, “Expectation-Maximization Attention Networks for Semantic Segmentation,” In Proceedings of the IEEE/CVF International Conference on Computer Vision, 9167–9176 (2019).

[32] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(6), 721–741 (1984).

[33] A. Amin, “Bayesian Inference for Double Seasonal Moving Average Models: A Gibbs Sampling Approach,” Pakistan Journal of Statistics and Operation Research, 483–499 (2017).

[34] J. Su, J. Xu, X. Qiu, and X. Huang, “Incorporating Discriminator in Sentence Generation: a Gibbs Sampling Method,” In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 5496–5503 (2018).

[35] Y. Huang and J. L. Beck, “Full Gibbs Sampling Procedure for Bayesian System Identification Incorporating Sparse Bayesian Learning with Automatic Relevance Determination,” Computer - Aided Civil and Infrastructure Engineering, 33(9), 712–730 (2018).

[36] G. G. Ko, Y. Chai, R. A. Rutenbar, D. Brooks, and G. Y. Wei, “Accelerating Bayesian Inference on Structured Graphs Using Parallel Gibbs Sampling,” In 29th International Conference on Field Programmable Logic and Applications, 159–165 (2019).

[37] H. Park, T. Park, and Y. S. Lee, “Partially Collapsed Gibbs Sampling for Latent Dirichlet Allocation,” Expert Systems with Applications, 131, 208–218 (2019).

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, “Equation of State Calculations by Fast Computing Machines,” The Journal of Chemical Physics, 21(6), 1087–1092 (1953).

[39] W. K. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” Biometrika 57(1), 97–109 (1970).

[40] N. Miao, H. Zhou, L. Mou, R. Yan, and L. Li, “CGMH: Constrained Sentence Generation by Metropolis-Hastings Sampling,” In Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 6834–6842 (2019).

[41] R. Turner, J. Hung, E. Frank, Y. Saatchi, and J. Yosinski, “Metropolis-Hastings Generative Adversarial Networks,” In International Conference on Machine Learning, 6345–6353 (2019).

[42] H. Park, S. Lee, S. Kim, J. Park, J. Jeong, K. M. Kim, J. W. Ha, and H. J. Kim, “Metropolis-Hastings Data Augmentation for Graph Neural Networks,” In Thirty-Fifth Conference on Neural Information Processing Systems (2021).

[43] N. Cunningham, J. E. Griffin, and D. L. Wild, “ParticleMDI: Particle Monte Carlo Methods for the Cluster Analysis of Multiple Datasets with Applications to Cancer Subtype Identification,” Advances in Data Analysis and Classification, 14(2), 463–484 (2020).

[44] M. Iloska, Y. El-Laham, and M. F. Bugallo, “A Particle Gibbs Sampling Approach to Topology Inference in Gene Regulatory Networks,” In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, 5855–5859 (2020)

[45] S. Wang, L. Wang, “Particle Gibbs Sampling for Bayesian Phylogenetic Inference,” Bioinformatics, 37(5), 642–649 (2021).

[46] A. Jasra, A. Persing, A. Beskos, K. Heine, M. D. Iorio, “Bayesian Inference for Duplication–Mutation with Complementarity Network Models,” Journal of Computational Biology, 22(11), 1025–1033 (2015).

[47] D. Du, Z. Hu, Y. Du, “Model Identification and Physical Exercise Control Using Nonlinear Heart Rate Model and Particle Filter,” IEEE 15th International Conference on Automation Science and Engineering, 405–410 (2019).

[48] K. K. Osmundsen, T. S. Kleppe, R. Liesenfeld, A. Oglend, “Estimating the Competitive Storage Model with Stochastic Trends in Commodity Prices,” Econometrics, 9(4), 40 (2021).

[49] K. Hukushima and K. Nemoto, “Exchange Monte Carlo Method and Application to Spin Glass Simulations,” Journal of the Physical Society of Japan, 65(6), 1604–1608 (1996).

[50] R. Ueno and Y. Okamoto, “Designed-Walk Replica-Exchange Method for Simulations of Complex Systems,” Computer Physics Communications, 196, 380–383 (2015).

[51] K. Motonaka, and S. Miyoshi, “Connecting PM and MAP in Bayesian Spectral Deconvolution by Extending Exchange Monte Carlo Method and Using Multiple Data Sets,” Neural Networks, 118, 159–166 (2019).

[52] A. L. Hodgkin and A. F. Huxley, “A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve,” The Journal of Physiology, 117(4), 500–544 (1952).

[53] R. Fitzhugh, “Impulses and Physiological States in Theoretical Models of Nerve Membrane,” Biophysical Journal, 1(6), 445–466 (1961).

[54] C. Morris and H. Lecar, “Voltage Oscillations in the Barnacle Giant Muscle Fiber,” Biophysical Journal, 35(1), 193–213 (1981).

[55] R. M. Rose and J. L. Hindmarsh, “The Assembly of Ionic Currents in a Thalamic Neuron I. The Three-Dimensional Model,” Proceedings of the Royal Society of London. B. Biological Sciences, 237(1288), 267–288 (1989).

[56] H. R. Wilson, “Simplified Dynamics of Human and Mammalian Neocortical Neurons,” Journal of Theoretical Biology, 200(4), 375–388 (1999).

[57] G. D. Smith, C. L. Cox, S. M. Sherman, and J. Rinzel, “Fourier Analysis of Sinusoidally Driven Thalamocortical Relay Neurons and a Minimal Integrate-and-Fire-or-Burst Model,” Journal of Neurophysiology, 83(1), 588–610 (2000).

[58] P. E. Latham, B. J. Richmond, P. G. Nelson, and S. Nirenberg, "Intrinsic Dynamics in Neuronal Networks. I. Theory," Journal of Neurophysiology, 83(2), 808–827 (2000).

[59] E. M. Izhikevich, "Resonate-and-Fire Neurons," Neural Networks, 14(6–7), 883–894 (2001).

[60] E. M. Izhikevich, "Simple Model of Spiking Neurons," IEEE Transactions on Neural Networks, 14(6), 1569–1572 (2003).

[61] E. M. Izhikevich, "Which Model to Use for Cortical Spiking Neurons?" IEEE Transactions on Neural Networks, 15(5), 1063–1070 (2004).

[62] Y. Dong, S. Mihalas, A. Russell, R. Etienne-Cummings, and E. Niebur, "Estimating Parameters of Generalized Integrate-and-Fire Neurons from the Maximum Likelihood of Spike Trains," Neural Computation, 23(11), 2833–2867 (2011).

[63] O. E. Barndorff-Nielsen, N. Shephard, "Non-Gaussian Ornstein–Uhlenbeck-Based Models and Some of Their Uses in Financial Economics," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 167–241 (2001).

[64] O. E. Barndorff-Nielsen, N. Shephard, "Normal Modified Stable Processes," Theory of Probability and Mathematical Statistics, 65, 1–19 (2002).

[65] I. SenGupta, W. Nganje, and E. Hanson, "Refinements of Barndorff-Nielsen and Shephard Model: an Analysis of Crude Oil Price with Machine Learning," Annals of Data Science, 8(1), 39–55 (2022).

List of Publications

Journal papers

1. H. Inoue, K. Hukushima, and T. Omori, “Replica Exchange Particle-Gibbs Method with Ancestor Sampling,” Journal of the Physical Society of Japan, 89(10), 104801:1–7 (2020).
2. H. Inoue, K. Hukushima, and T. Omori, “Estimation of Neuronal Dynamics of Izhikevich Neuron Models from Spike-Train Data with Particle Markov Chain Monte Carlo Method,” Journal of the Physical Society of Japan, 90(10), 104801:1–9 (2021).
3. H. Inoue, K. Hukushima, and T. Omori, “Estimating Distributions of Parameters in Nonlinear State Space Models with Replica Exchange Particle Marginal Metropolis-Hastings,” Entropy, 24(1), 115:1–20 (2022).

International conferences

1. H. Inoue and T. Omori, “Bayesian Probabilistic Approach for Estimating Neural System Based on Calcium Imaging,” Cyber-Physical System for Smarter World (2016).
2. H. Inoue and T. Omori, “Statistical Estimation of Neural System Using Calcium Imaging,” Proceedings of the 16th International Symposium on Advanced Intelligent Systems 2015, 819–826 (2015).

3. H. Inoue and T. Omori, “Bayesian Estimation of Neural Systems Using Particle-Gibbs,” ACM International Conference Proceedings Series, 68–73 (2017).
4. H. Inoue, K. Hukushima and T. Omori, “Estimation of Neural Dynamics with Particle Markov Chain Monte Carlo,” The 9th RIEC International Symposium on Brain Functions and Brain Computer (2020).
5. H. Inoue, T. Omori, “Data-Driven Method for Estimating Neuronal Nonlinear Dynamics from Noisy Partial Observation,” Proceedings of the 27th International Symposium on Artificial Life and Robotics, 1–6 (2022).

Doctor Dissertation, Kobe University
“Estimating Latent Dynamics for Time-Series Data with Particle Markov Chain Monte Carlo Methods”, 100 pages
Submitted on January, 21st, 2022
The date of publication is printed in cover of repository version published in Kobe University Repository Kernel.