A Synthesis Method of Spiking Neural Oscillators with Considering Asymptotic Stability

著者	Kuroe Yasuaki, Miyoshi Seiji, Hikawa Hiroomi, Ito Hidetaka, Motonaka Kimiko, Maeda Yutaka
journal or publication title	2021 International Joint Conference on Neural Networks (IJCNN) Proceedings
会議概要（会議名, 開催地, 会期, 主催者等）	Date of Conference: 18-22 July 2021 Conference Location: Shenzhen, China
権利	(C) 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
URL	http://hdl.handle.net/10112/00025440

doi: 10.1109/IJCNN52387.2021.9534255

A Synthesis Method of Spiking Neural Oscillators with Considering Asymptotic Stability

Yasuaki Kuroe, Seiji Miyoshi, Hiroomi Hikawa, Hidetaka Ito, Kimiko Motonaka and Yutaka Maeda
Faculty of Engineering Science
Kansai University
Suita, Japan
kuroe@ieee.org, {miyoshi, hikawa, h.ito, motonaka, maedayut}@kansai-u.ac.jp

***Abstract*—In artificial Spiking Neural Networks (SNNs) the information processing and transmission are carried out by spike trains in a manner similar to the generic biological neurons. Recently it has been reported that they are computationally more powerful than the conventional neural networks. In biological systems there are numerous examples of autonomously generated periodic activities. Several different periodic patterns are generated simultaneously in a living body. It is known that in biological systems there are specific neurons which generate such periodic patterns. This paper presents a method for synthesis of neural oscillators by spiking neural networks. We propose a learning method for synthesizing spiking neural networks which generate desired periodic spike trains with specified spike emission times. We also propose a method for making the periodic trajectory generated by the synthesized spiking neural oscillator asymptotically stable.**

***Index Terms*—neural oscillator, spiking neural network, synthesis method, stability**

I. Introduction

In biological systems there are numerous examples of autonomously generated periodic motor activities, such as locomotion, mastication, respiration and so on. It is known that in biological systems there are specific neurons which generate such periodic patterns. In engineering applications, there are a lot of problems which require to generate periodic patterns such as repetitive motion control of robots. Owing to these reasons, several studies on artificial neural oscillators have been done [1]–[3], and most of them use conventional threshold or sigmoidal neural networks. In this paper we present a synthesis method of artificial neural oscillator by using spiking neural networks.

In last decades there is a surge in the research of artificial spiking neural networks (SNNs) due to the fact that the functions of spiking neurons are closer to the physiological functions of the generic biological neurons than the conventional threshold and sigmoidal neurons [4]–[6]. In artificial spiking neural networks the information is encoded and processed by the spike trains (sequence of action potentials) similar to the biological neural networks (BNNs) through a discontinuous and nonlinear encoding mechanism [4], [5]. The conventional neuron models usually tend to ignore these sophisticated discontinuous encoding mechanisms. In addition to the SNNs' similarity to the BNNs, recently it has been reported that they are computationally more powerful than the conventional artificial neural networks [6]–[8]. It is, however, much more difficult to analyze and synthesize the SNNs than the conventional threshold and sigmoidal neural networks. This is due to their nonlinear and discontinuous encoding mechanisms, which make the SNNs continuous and discrete hybrid-dynamical-systems.

We have already proposed a learning method for synthesizing artificial neural oscillators using spiking neural networks [9]. The method makes it possible to synthesize a spiking neural network which generate a desired periodic trajectory. However, the method does not guarantee stability of generated periodic trajectories, which is an important problem, especially for engineering applications. In this paper we propose a synthesis method of spiking neural oscillators with ensuring asymptotic stability of the generated periodic trajectory. The proposed synthesis method is based on learning of neural networks and it makes possible to synthesize a spiking neural network which generates a desired periodic trajectory with asymptotic stability. It is known that the stability of periodic trajectories can be investigated by checking eigenvalues of Jacobian matrix of the Poincaré map defined on them. In order to make the generated periodic trajectory asymptotically stable, we propose a learning method of neural networks such that Jacobian matrix of the Poincaré map of generated periodic trajectory possess specified stable eigenvalues. We have implemented the proposed synthesis method by using the simulator of the SNN. Numerical experiments are carried out to check the performance of the proposed method. It is shown that the proposed method makes it possible to realize the spiking neural oscillators which can generate desired asymptotically stable periodic trajectories.

II. Spiking Neural Networks

A. Firing Mechanism of Integrate and Fire Type Spiking Neurons

In this paper we consider recurrent spiking neural networks in which integrate-and-fire type spiking neurons (SNs) are fully connected. The firing mechanism of the i^{th} integrate-and-fire type spiking neuron in the network is shown in Fig.1. When an input stimulus $e_i(t)$ is fed into the integrator with

This is a product of research which was financially supported in part by the Kansai University Fund for the Promotion and Enhancement of Education and Research, 2018, "Development of System Analysis and Design Methods Integrating Model and Data Premised on IOT Technology".

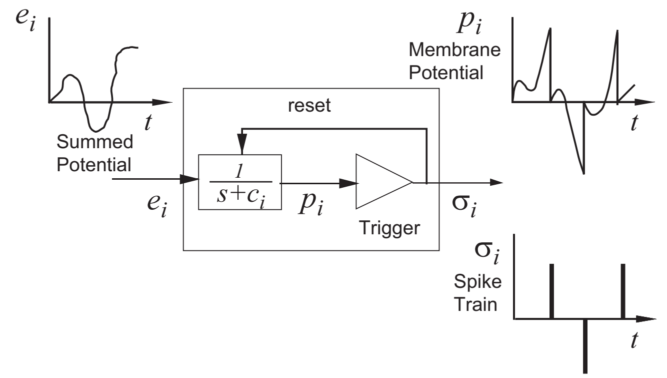

Fig. 1. Schematic of the firing mechanism of the integrate and fire type SN.

transfer function $1/(s+c_i)$ (timing filter), a spike is emitted at the moment when the filter-output $p_i(t)$ reaches the threshold value s_i. At the instant of spike emission the sign of the filter output $p_i(t)$ is observed and assigned to the output spike and the internal states of the filter are reset to zero. The firing mechanism is mathematically described as follows.

$$\sigma_i(t) = \sum_{k_i=1}^{K_i} \varepsilon_{i,k_i} \times \delta(t - t_{i,k_i}) \tag{1}$$

$$t_{i,k_i} = \min[t : t > t_{i,k_i-1}, |p_i(t)| \geq s_i] \tag{2}$$

$$\varepsilon_{i,k_i} = \mathrm{sgn}[p_i(t^-_{i,k_i})] \tag{3}$$

$$\frac{dp_i(t)}{dt} = -c_i p_i(t) + e_i(t), \quad t_{i,k_i-1} < t < t_{i,k_i} \tag{4}$$

$$p_i(0) = p_i^0, \tag{5}$$

$$p_i(t^+_{i,k_i}) = 0, \quad k_i = 1, \ldots K_i, \tag{6}$$

where, $\sigma_i(t)$: output sequence of spikes of the i^{th} spiking neuron, K_i: total number of spikes fired before time t, t_{i,k_i}: time at which the k_i^{th} spike is emitted, $p_i(t)$: output of the filter, p_i^0: initial condition of the filter, $1/c_i$: time constant, s_i: threshold value, $e_i(t)$: input to the spiking neuron, and $p_i(t^-_{i,k_i})$ = $\lim_{\varepsilon\to 0} p_i(t_{i,k_i} - \varepsilon)$, $p_i(t^+_{i,k_i})$ = $\lim_{\varepsilon\to 0} p_i(t_{i,k_i} + \varepsilon)$, $\varepsilon > 0$. Equation (6) represents the resetting mechanism of the SN at spike emission times.

B. Model of Spiking Neural Networks

We consider recurrent spiking neural networks in which integrate-and-fire type spiking neurons shown in Fig. 1 are fully connected through synaptic weights $w_{i,j}$ and time delay elements $g_{i,j}(s)$. Figure 2 shows a schematic diagram of the connection of the ith spiking neuron in the recurrent SNN. We let here the number of neurons composed in the recurrent SNN be M. The elements $g_{i,j}(s)$ determine shape of post synaptic potentials, so called spike-response function as shown in Fig. 3, or delay due to the spike transmission between spiking neurons. Synaptic weights $w_{i,j}$ are added to the time delay

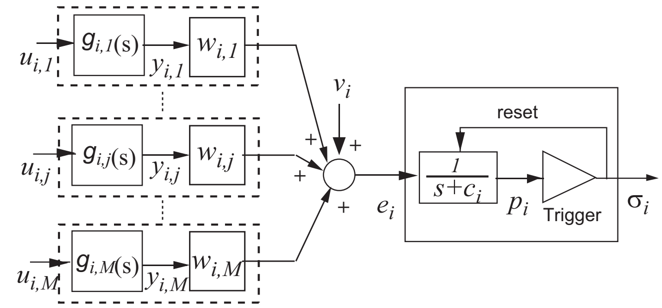

Fig. 2. Model of the connections of ith spiking neuron in the SNN.

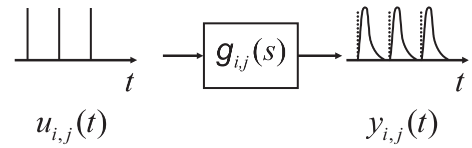

Fig. 3. Spikes and spike-response function.

elements ($w_{i,j} \times g_{i,j}(s)$). A typical example of the spike-response function is a difference of two exponentially decaying functions and the elements $g_{i,j}(s)$ is given by

$$g_{i,j}(s) = \kappa \left\{ \frac{1}{s + 1/\tau_m} - \frac{1}{s + 1/\tau_s} \right\}, 0 < \tau_s < \tau_m,\ \kappa > 0.$$

The input stimulus $e_i(t)$ of the i^{th} SN is generated by the weighted sum of each output $y_{i,j}(t)$ of the element $g_{i,j}(s)$, and the external input $v_i(t)$. The input $u_{i,j}(t)$ of $g_{i,j}(s)$ is connected with the output of the j^{th} neuron, $\sigma_j(t)$. The connection topology of the entire SNN is given by

$$e_i(t) = \sum_{j=1}^{M} w_{i,j} y_{i,j}(t) + v_i(t), \tag{7}$$

$$u_{i,j}(t) = \sigma_j(t), \tag{8}$$

For the convenience of the derivation of the learning algorithms, the elements $g_{i,j}(s)$ are expressed in state space form:

$$\frac{d\boldsymbol{x}_{i,j}(t)}{dt} = \boldsymbol{A}_{i,j}\boldsymbol{x}_{i,j}(t) + \boldsymbol{b}_{i,j}u_{i,j}(t) \tag{9}$$

$$y_{i,j}(t) = \boldsymbol{c}_{i,j}\boldsymbol{x}_{i,j}(t) \tag{10}$$

$$\boldsymbol{x}_{i,j}(0) = \boldsymbol{x}^0_{i,j}, \qquad i, j = 1, \cdots, M \tag{11}$$

$$g_{i,j}(s) = \boldsymbol{c}_{i,j}(sI - \boldsymbol{A}_{i,j})^{-1}\boldsymbol{b}_{i,j},$$

where $\boldsymbol{x}_{i,j}(t)$: N dimensional state vector, $\boldsymbol{x}^0_{i,j}$: N dimensional initial state vector and, the dimensions of the system matrices and vectors $\boldsymbol{A}_{i,j}$, $\boldsymbol{b}_{i,j}$ and $\boldsymbol{c}_{i,j}$ are $N \times N$, $N \times 1$ and $1 \times N$, respectively. Equations (1)∼(11) give the whole description of the recurrent SNN considered in this paper.

III. Proposed Synthesis Method

A. Synthesis Method of Spiking Neural Oscillators Generating Desired Spike Trains

In this section we discuss a synthesis method of spiking neural oscillators by using the spiking neural network described by (1)∼(11). The objective here is to synthesize a spiking neural oscillator such that the spiking neural network possesses an autonomous persistent spike-train oscillation with a given period, and given number of spikes at given firing instants. Let T be a desired period of the oscillation, K_i^d be a desired number of spikes within the period, t_{i,k_i}^d $(0 < t_{i,1}^d < t_{i,2}^d \ldots < t_{i,K_i^d}^d < T)$ be desired spike firing instants and ε_{i,k_i}^d be desired sign of the spikes. The objective is to find the values of the synaptic weights $w_{i,j}$, which realize that in the steady state the actual firing times t_{i,k_i}^a and actual sign of ε_{i,k_i}^a of each spiking neuron coincide with the given desired spike firing times t_{i,k_i}^d,

$$t_{i,k_i}^a = t_{i,k_i}^d, \quad k_i = 1, 2, \ldots, K_i^d, \quad i = 1, 2, \ldots, M \tag{12}$$

and the given desired sign of the spikes, $\varepsilon_{i,k_i}^a = \varepsilon_{i,k_i}^d$, and satisfy the periodicity condition,

$$\sigma_i(t) = \sigma_i(t + T). \tag{13}$$

Furthermore, since all the states of the spiking neural network need be periodic, the following conditions are required.

$$\begin{aligned} \boldsymbol{x}_{i,j}(t) &= \boldsymbol{x}_{i,j}(t+T),\ p_i(t) = p_i(t+T), \\ & (i, j = 1, 2 \ldots, M). \end{aligned} \tag{14}$$

The objective here is to find values of the synaptic weights $w_{i,j}$ such that (12), (13) and (14) are satisfied.

In order to realize the relationships (12), (13) and (14), we define the following cost function.

$$J_1 = \frac{1}{2}\left(\alpha \sum_{i=1}^{M} \sum_{k_i=1}^{K_i^d} (t_{i,k_i}^d - t_{i,k_i}^a)^2 + \beta \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{n=1}^{N} (x_{i,j}^{0,n} - x_{i,j}^n(T))^2 + \gamma \sum_{i=1}^{M} (p_i^0 - p_i(T))^2 \right), \tag{15}$$

where $x_{i,j}^{0,n} := x_{i,j}^n(0)$, and α, β and γ are suitable positive weighting coefficients. Note that in the spiking neural network, the initial conditions which generate autonomous oscillatory responses are unknown. Therefore it becomes necessary to treat the initial conditions as unknown variables. The objective is now reduced to solving the following constrained optimization problem.

$$\begin{aligned} \text{Minimize} \quad & J_1 \\ \text{w.r.t :} \quad & w_{i,j}, \boldsymbol{x}_{i,j}^0, p_i^0 \ (i, j = 1, 2, \ldots, M) \\ \text{subject to :} \quad & K_i^d = K_i^a, \\ & |p_i^0| < s_i \ (i = 1, 2, \ldots, M), \end{aligned} \tag{16}$$

where K_i^a is the actual number of spikes fired by the ith spiking neuron within the period T.

B. Synthesis Method for Making Generated Periodic Trajectory Asymptotically Stable

It should be noted that the synthesis method explained above does not guarantee that the realized periodic spike trains are stable periodic trajectories. In this section we propose a method for making the realized periodic spike trains asymptotically stable periodic trajectories. It is known that an important tool for investigating the stability of a periodic solution of a dynamical system is the Poincaré map, which maps the initial values of the system state vector to those one period T later [10]. Let $\boldsymbol{z}$ be the state vector of the spiking neural network, defined by

$$\boldsymbol{z} = [p_1, p_2, \cdots, p_M, \boldsymbol{x}_{1,1}^T, \boldsymbol{x}_{1,2}^T, \cdots, \boldsymbol{x}_{M,M}^T]^T \in R^{N_s} \tag{17}$$

where $N_s = M + N \times M \times M$. Note that $\boldsymbol{z}(T)$ is a function, denoted by $P(\cdot)$, of an initial state $\boldsymbol{z}_0 := \boldsymbol{z}(0)$:

$$\boldsymbol{z}(T) = P(\boldsymbol{z}_0). \tag{18}$$

The function $P(\cdot)$ corresponds to the Poincaré map. Assume that the spiking neural network has a periodic trajectory with period T, and we denote the periodic trajectory by $\boldsymbol{z}^*(t)$ which satisfies the periodic condition:

$$\boldsymbol{z}^*(T) = P(\boldsymbol{z}^*(0)) = \boldsymbol{z}^*(0).$$

This means that a point in the periodic trajectory $\boldsymbol{z}^*(t)$ is a fixed point of the Poincaré map. Therefore the stability analysis of the periodic solution $\boldsymbol{z}^*(t)$ is reduced to that of the fixed point of the Poincaré map $P(\cdot)$ and it is known the following fact on the stability analysis.

Let $DP(\boldsymbol{z}) \in R^{(N_s)\times(N_s)}$ be the Jacobian matrix of the Poincaré map $P(\cdot)$ with respect to $\boldsymbol{z}$, defined by

$$DP(\boldsymbol{z}) = \frac{\partial P(\boldsymbol{z})}{\partial \boldsymbol{z}}, \tag{19}$$

and let $DP(\boldsymbol{z}^*(0))$ be the the Jacobian matrix of the Poincaré map $P(\cdot)$ at the point $\boldsymbol{z}^*(0)$ of the periodic trajectory $\boldsymbol{z}^*(t)$. It is known that the following fact on the asymptotic stability holds [10]. The Jacobian matrix $DP(\boldsymbol{z}^*(0))$ always has one eigenvalue of unity and if the absolute values of the $N_s - 1$ other eigenvalues of $DP(\boldsymbol{z}^*(0))$ are all less than unity (i.e. are all inside the unit circle with its center being at the origin in the complex plane), then the periodic trajectory $\boldsymbol{z}^*(t)$ is asymptotically stable. By using this fact a synthesis method which makes the generated periodic trajectory asymptotically stable can be obtained as follows.

In order to make the generated periodic trajectory asymptotically stable, we propose a method for assigning all the eigenvalues of the Jacobian matrix $DP(\boldsymbol{z}^*(0))$ of the Poincaré map $P(\cdot)$ to be at desired positions in the complex plane [11]. Let

$$\boldsymbol{\mu}^{\boldsymbol{z}^*} = \{\mu_1^{\boldsymbol{z}^*}, \mu_2^{\boldsymbol{z}^*}, \cdots, \mu_{N_s}^{\boldsymbol{z}^*}\}$$

be a set of desired eigenvalues which are assigned to the matrix $DP(\boldsymbol{z}^*(0))$. Note that if $\mu_i^{\boldsymbol{z}^*}$ is a complex number, its complex conjugate $\bar{\mu}_i^{\boldsymbol{z}^*}$ should also be in $\boldsymbol{\mu}^{\boldsymbol{z}^*}$, and one eigenvalue of unity should be in $\boldsymbol{\mu}^{\boldsymbol{z}^*}$. In the following, for the sake of simplicity, $DP(\boldsymbol{z}^*(0))$ will be abbreviated as $\boldsymbol{D}^{z^*}$ ($\boldsymbol{D}^{z^*} := DP(\boldsymbol{z}^*(0))$). By using the Leverrier-Bôcher formula [12], the Jacobian matrix $\boldsymbol{D}^{z^*}$ has $\boldsymbol{\mu}^{z^*} = \{\mu_1^{z^*}, \mu_2^{z^*}, \cdots, \mu_{N_s}^{z^*}\}$ as its eigenvalues if the following relations hold:

$$\beta_{N_s-i}^{z^*} = -\frac{1}{i}\{tr[(\boldsymbol{D}^{z^*})^i] + \beta_{N_s-1}^{z^*} tr[(\boldsymbol{D}^{z^*})^{i-1}] + \cdots + \beta_{N_s-i+1}^{z^*} tr[\boldsymbol{D}^{z^*}]\} \ \ (i = 1, 2, \cdots, N_s) \quad (20)$$

where $\beta_i^{z^*}$, $i = 1, 2, \cdots, N_s$ satisfy

$$\prod_{i=1}^{N_s}(s - \mu_i^{z^*}) = \beta_0^{z^*} + \beta_1^{z^*} s + \cdots + \beta_{N_s-1}^{z^*} s^{N_s-1} + s^{N_s}. \quad (21)$$

Now we are led to define the following cost function J_2:

$$J_2 = \frac{1}{2}\sum_{i=1}^{N_s}\{h(\boldsymbol{D}^{z^*}, \beta_{i-1}^{z^*})\}^2 \quad (22)$$

where

$$h(\boldsymbol{D}^{z^*}, \beta_{i-1}^{z^*}) = -\frac{1}{i}\{tr[(\boldsymbol{D}^{z^*})^i] + \beta_{N_s-1}^{z^*} tr[(\boldsymbol{D}^{z^*})^{i-1}] \cdots + \beta_{N_s-i+1}^{z^*} tr[\boldsymbol{D}^{z^*}]\} - \beta_{N_s-i}^{z^*}. \quad (23)$$

Thus the problem is reduced to finding the parameters such as $w_{i,j}$, c_i, $\boldsymbol{A}_{i,j}$, $\boldsymbol{b}_{ij}$ and $\boldsymbol{c}_{ij}$ of the spiking neural network which minimize J_2. If this minimization succeeds and the cost function can be reduced to zero, the Jacobian matrix $DP(\boldsymbol{z}^*(0))$ of the Poincaré map $P(\cdot)$ will possess the specified eigenvalues $\boldsymbol{\mu}^{z^*}$. Note that the network parameters do not always exist such that the eigenvalues of the Jacobi matrix of the Poincaré map become equal to the specified eigenvalues $\boldsymbol{\mu}^{z^*}$, in which case the optimization does not succeed completely. It is expected, however, that the eigenvalues of the Jacobian matrix of the Poincaré map will settle inside the unit circle and the generated periodic trajectory becomes asymptotically stable if the absolute values of the desired eigenvalues are chosen small enough and the objective function can be reduced small enough by the optimization.

C. Learning Method

We have introduced two cost functions (15) and (22). In order to minimize those two cost functions simultaneously we define the following augmented cost function:

$$J = J_1 + \delta J_2 \quad (24)$$

where $\delta > 0$ is a weighting coefficient. The objective now is reduced to solving the following optimization problem.

$$\begin{aligned} &\text{Minimize} && J \quad (25)\\ &\text{w.r.t :} && w_{i,j}, \boldsymbol{x}_{i,j}^0, p_i^0, \boldsymbol{q}_{i,j} \ (i, j = 1, 2, \ldots, M)\\ &\text{subject to :} && K_i^d = K_i^a,\\ & && |p_i^0| < s_i \ (i = 1, 2, \ldots, M), \end{aligned}$$

where $\boldsymbol{q}_{i,j}$ is the vector consisting of the parameters of the spiking neural network such as c_i, $\boldsymbol{A}_{i,j}$, $\boldsymbol{b}_{i,j}$ and $\boldsymbol{c}_{i,j}$. In order to solve this optimization problem we use gradient based methods, in which several useful algorithms are available: the steepest decent algorithm, the conjugate gradient algorithm, the quasi-Newton algorithm so on. Main problem associated with these algorithms is the computation of the gradients of the cost function J with respect to the decision variables $w_{i,j}$, $\boldsymbol{x}_{i,j}^0$, p_i^0, $\boldsymbol{A}_{i,j}$, $\boldsymbol{b}_{i,j}$ and $\boldsymbol{c}_{i,j}$. An efficient algorithm to compute those gradients can be derived by introducing the adjoint networks to the spiking neural network [14]. The derivation of the gradients is omitted here.

IV. Synthesis Example

We have implemented the proposed synthesis method by using the simulator of the spiking neural network. In this section we present a synthesis example to demonstrate the applicability and performance of the proposed method. We use a spiking neural network constructed with two spiking neurons as shown in Fig.4. This spiking neural network has 10 state variables p_1, p_2, $\boldsymbol{x}_{ij} \in R^2$ $(i, j = 1, 2)$, and Jacobian matrix $DP(\boldsymbol{z}^*(0))$ of the Poincaré map as $\boldsymbol{\mu}^{z^*}$ is the 10×10 matrix. In solving the optimization problem (25) we choose the weighting coefficients as $\alpha = \beta = \gamma = 1$, $\delta = 10^{-5}$ and $\boldsymbol{q}_{i,j}$ as the vector consisting of c_1, $\boldsymbol{A}_{i,j}$. The other parameters of the spiking neural network are chosen as: the threshold value $s_i = 0.8$, parameters of delay elements $\boldsymbol{b}_{ij} = [1, 1]^T$, $\boldsymbol{c}_{ij} = [1, -1]$. By using the proposed learning method, we synthesize the spiking neural oscillator in such a way that it possesses the autonomous periodic spike train with the period $T = 2.0$, the number of pulses within one period $K_i^d = 2$, and the spike firing times $t_{i,1}^d = 0.5$ and $t_{i,2}^d = 1.5$, $i = 1, 2$. In order to make the generated periodic trajectory asymptotically stable, we choose the desired eigenvalues $\boldsymbol{\mu}^{z^*}$ which are assigned to the Jacobian matrix $DP(\boldsymbol{z}^*(0))$ of the Poincaré map as $\boldsymbol{\mu}^{z^*} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$. Note that, in the assigned eigenvalues, one eigenvalue is unity and 9 other eigenvalues are zero, that is, are all inside the unit circle with its center being at the origin in the complex plane.

We used the quasi-Newton algorithm (the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method) to solve the optimization problem (25). Figure 5 shows an example of convergence behavior of the proposed leaning method, where values of the cost functions J versus the number of the learning iterations are plotted. It is observed that the proposed learning method converges quickly. Table I shows the firing instants of the spike trains generated by the synthesized spiking neural network, in which the desired firing instants are also shown for comparison. It can be seen from the table that the proposed synthesis method successfully realizes the spike trains with the desired firing instants. The time evolution of the state variables p_1 and p_2 of the synthesized spiking neural network for 5 periods (from $t = 0$ to $t = 5T$) is shown in Fig. 6. Figure 7 shows the trajectory of the synthesized spiking neural network in the state space. The dimension of the state space of the spiking neural network shown in Fig 4 is ten and the trajectory shown in Fig. 7 is the one projected on to the three dimensional state space ($x_{1,1}^1$, $x_{1,1}^2$, p_1). It is observed from Figs 6 and 7 that the trajectory is periodic with the period

$T = 2$ and synthesis method successfully realizes the desired spiking neural oscillator.

We evaluated the eigenvalues of the Jacobian matrix of the Poincaré map for the periodic trajectory of the synthesize spiking neural oscillator. Table II shows the values of the obtained eigenvalues and their absolute values, and Fig. 8 shows their positions in the complex plane. It is observed that one eigenvalue is unity and other eigenvalues are all inside the unit circle with its center being at the origin in the complex plane. Therefore it can be said the proposed method successfully make the periodic trajectory generated by the synthesized spiking neural oscillator asymptotically stable. To verify the stability, we solve the model equations of the synthesized spiking neural oscillator with the initial condition being perturbed from the periodic trajectory. The result is shown in Figs 9 and 10; the former is the time evolution of the state variables p_1 and p_2 and the latter is the trajectory in the state space $(x^1_{1,1}, x^2_{1,1}, p_1)$. It can be seen that they converge to the generated periodic trajectory, which imply that generated periodic trajectory is asymptotically stable.

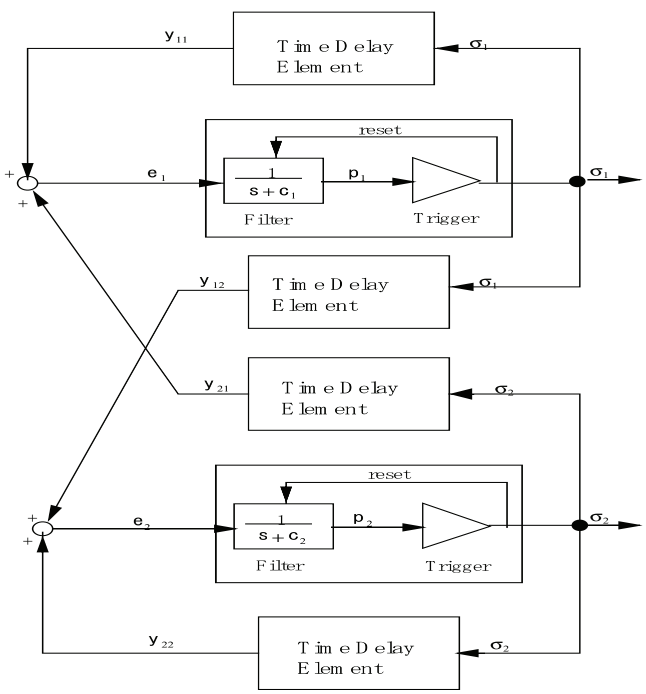

Fig. 4. A spiking neural network constructed with two spiking neurons.

TABLE I
COMPARISON BETWEEN REALIZED FIRING INSTANTS OF THE SPIKING NEURAL NETWORK SYNTHESIZED BY THE PROPOSED METHOD AND THEIR DESIRED ONES.

		$t_{i,1}$	$t_{i,2}$
Desired	SN_1,SN_2	0.50	1.50
Realized	SN_1	0.5000241	1.499966
Realized	SN_2	0.4999817	1.500025

TABLE II
THE OBTAINED EIGENVALUES OF THE JACOBIAN MATRIX OF THE POINCARÉ MAP FOR THE PERIODIC TRAJECTORY OF THE SYNTHESIZE SPIKING NEURAL OSCILLATOR

Obtained Eigenvalues	Absolute Values
$1.00000 + 0.00000i$	1.00000
$0.88756 + 0.00000i$	0.88756
$0.09693 + 0.57869i$	0.58675
$0.09693 - 0.57869i$	0.58675
$-0.37842 + 0.19385i$	0.42518
$-0.37842 - 0.19385i$	0.42518
$-0.12704 + 0.22139i$	0.25525
$-0.12704 - 0.22139i$	0.25525
$0.00438 + 0.00000i$	0.00438
$0.00504 + 0.00000i$	0.00504

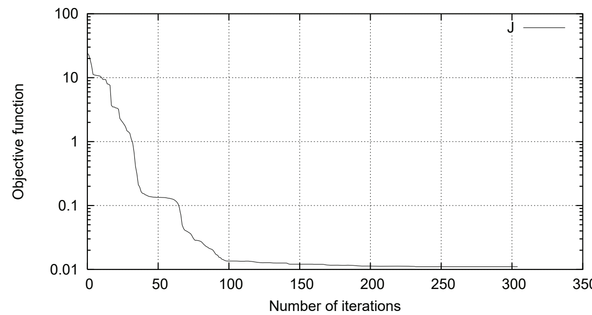

Fig. 5. An example of convergence behavior of the objective function during learning.

V. CONCLUSION

In this paper we have presented a synthesis method of neural oscillators by spiking neural networks. Fully connected recurrent spiking neural networks constructed with integrate-and-fire type spiking neurons have been considered. We have proposed a learning method which can determine values of parameters of the spiking neural networks such that they generate the desired periodic spike trains with the specified

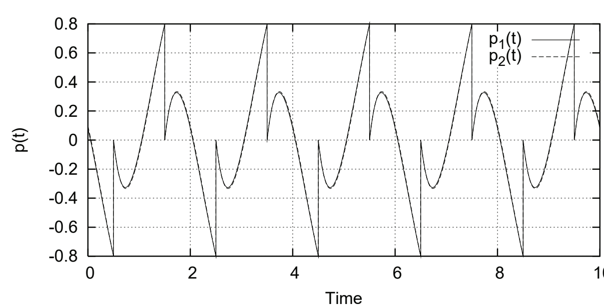

Fig. 6. The time evolution of the state variables p_1 and p_2 of the synthesized spiking neural oscillator

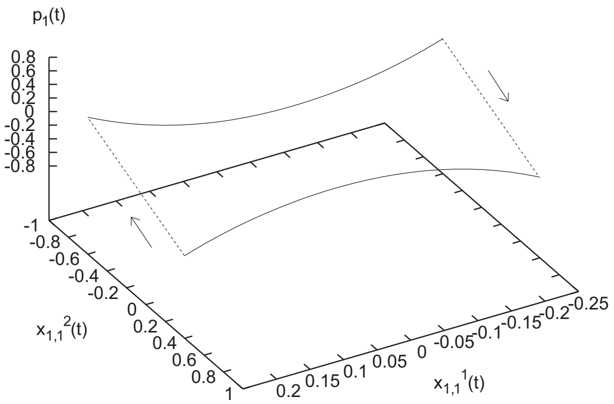

Fig. 7. The periodic trajectory of the synthesized spiking neural oscillator in the state space ($x^1_{1,1}$, $x^2_{1,1}$, p_1)

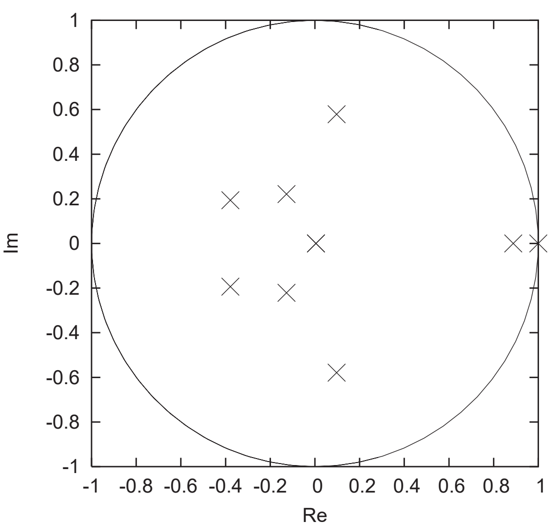

Fig. 8. The positions of the obtained eigenvalues of the Jacobian matrix of the Poincaré map in the complex plane.

spike emission instants. A learning method for making the generated periodic trajectory asymptotically stable has been also proposed. Synthesis examples have been provided to show the validity and performance of the proposed method.

The authors wish to express their gratitude to Tomokazu Ueyama for his valuable discussions.

REFERENCES

[1] K.Matsuoka, “Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators,” Biological Cybernetics, Vol.56, pp.345-353, 1987.

[2] H.Kimura, K.Sakurama and S.Akiyama, “Dynamic Walking and Running of the Quadruped Using Neural Oscillator,” Proc. of IEEE&RSJ IROS98, Victoria, pp.50-57, 1998.

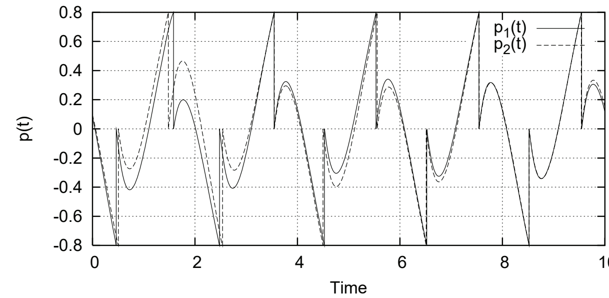

Fig. 9. The time evolution of the state variables p_1 and p_2 of the the synthesized spiking neural oscillator with the initial condition being perturbed

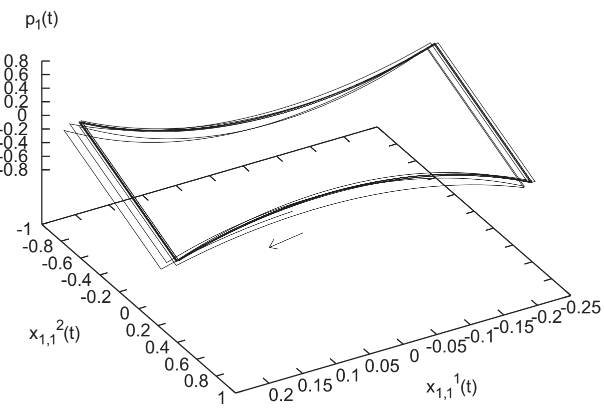

Fig. 10. The trajectory of the synthesized spiking neural oscillator with the initial condition being perturbed in the phase plane ($x^1_{1,1}$, $x^2_{1,1}$, P_1)

[3] K.Doya and S.Yoshizawa, “Adaptive Neural Oscillator Using Continuous-Time Back Propagation Learning,” Neural Networks, Vol.2, pp.375-385, 1989.

[4] W. Mass and C. Bishop, (Eds.), Pulsed Neural Nets, MIT Press, 1998.

[5] W. Gerstner and J. L. van Hemmen: “How to Describe Neural Activities: Spikes, Rates or Assemblies?,” Advances in Neural Information Processing Systems, San Mateo, Vol.6, pp.363-374, 1993.

[6] W. Mass, “Networks of Spiking Neurons: The Third Generation of Neural Network Models,” Neural Networks, Vol.10, No.9, pp.1659-1671, 1997.

[7] W. Mass, “Fast Sigmoidal Networks via Spiking Neurons,” Neural Computation, Vol.9, pp.279-304, 1997.

[8] W. Mass, “On the Computational Complexity of Network of Spiking Neurons,” Advances in Neural Information Processing Systems, MIT Press, Cambridge, Vol.7, pp.183-190, 1996.

[9] Y. Kuroe and H. Iima, “A Learning Method for Synthesizing Spiking Neural Oscillators,” Proc. of International Joint Conference on Neural Networks, pp.7613-7617, 2006.

[10] J. Guckenheimer and P. Holms, Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Field, New York: Springer-Verlag, 1983.

[11] Y. Kuroe, K. Koashi, N. Hashimoto and T. Mori, “A Learning Method for Synthesizing Associative Memory in Neural Networks,” Proc. of IEEE International Joint Conference on Neural Networks, #2028, pp.1-6, 1999.

[12] M. Bôcher, Introduction to Higher Algebra, Dover, New York, 1964.

[13] D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, pp.194-197, 1973.

[14] Y. Kuroe and T. Ueyama, “Learning Methods of Recurrent Spiking Neural Networks Based on Adjoint Equations Approach,” Proc. of WCCI 2010 IEEE World Congress on Computational Intelligence, pp.2561-2568, 2010.