Configuring Spiking Neural Networks for Given Spatio-Temporal Patterns

Takaki Makino Jianfeng Feng

Abstract

We have developed a general framework to configure a spiking neuronal network so that it can precisely generate a desired spatio-temporal pattern of spikes. The unit of spiking neuronal networks employed here is a leaky integrate-and-fire model with a connection delay. We have shown that the required network configuration, and its existence, can be characterized by a set of linear inequalities constructed from the given pattern. The stability of the configured spiking neuronal network is discussed, which lead us to applying some routine methods in linear-programming to solve the set of inequalities, and yields the desirable spiking neural network configuration. To demonstrate the application of our approach, numerical examples with randomly generated patterns were explored and included.

Keywords: Spiking neurons, Spatio-temporal pattern, Learning, Leaky integrate-and-fire

Takaki Makino is with Department of Frontier Science and Science Integration, University of Tokyo. E-mail: mak@scint.dpc.u-tokyo.ac.jp

Jianfeng Feng is with the Centre for Scientific Computing and Computer Science, Warwick University.

Configuring Spiking Neural Networks for Given Spatio-Temporal Patterns

I. Introduction

Almost all biological neurons emit and receive spikes; hence, spikes are essentially the information carriers. It is natural to expect that artificial neuronal networks should and have to employ a similar strategy: using spikes as functional units, or more generally, spatio-temporal patterns of spikes as computational and functionally meaningful units, if we attempt to reveal the underlying computational mechanisms of the nervous systems [1], [2], [3]. Spiking neuronal networks have been widely investigated in the past decades, both in theory and through experimentation. There is, however, one key issue which has not been resolved that hinders further development of spiking neuronal networks: to clearly demonstrate the advantage of a spiking neuronal network over the classical artificial neuronal networks in information processing. One of the difficulties in fulfilling such a task lies in the fact that we do not have a general purpose learning rule. In order to ensure that a spiking network can produce a series of desired spatio-temporal spiking patterns, such a rule is needed, similar to the back-propagation learning rule in the classical artificial neural networks.

We have been, along with many others, interested in this issue for many years [4], [5], [6]. It has turned out to be a difficult task to find a learning rule based on a cost function; for example, the distance between the current spike pattern and the desired, given spike pattern. If a leaky integrate-and-fire model is used, then we face the problem that the trajectory of its membrane potential is not differentiable and hence a gradient descent method is not directly applicable (see for example, [4]). If a more complex model such as Hodgkin-Huxley is used, the cost function becomes too difficult to handle. Some success has been reported in the literature for a simplified neuronal model, i.e. the phase model [7, and references therein].

In our current study, we adopt a different approach from the examples above. Our approach involves a one-step configuration process of a neural network that produces the desired firing pattern. First, we present a theoretical framework of the necessary conditions for a network to produce the desired firing pattern for an appropriate initial condition. The units of the network are leaky integrate-and-fire neurons with constant inputs. Our results show that increasing the number of neurons increases the freedom of the network configuration. In other words, with a large enough number of neurons and a given spiking pattern, multiple configuration are generally available. An analysis on the stability of the pattern generation enables us to propose an algorithm to select an optimally stable solution out of many (infinite) solutions, with the help of some routine methods developed in linear-programming. Numerical examples demonstrate that our approach can produce complex spatio-temporal patterns of spikes as desired.

As mentioned above, to the best of our knowledge, our approach is the first to successfully present a theoretical framework to yield a configuration of a spiking neural network for a desired, given pattern. This is one of our series of papers aiming to elucidate how to develop and implement learning rules for spiking neuronal networks.

We hope this will provide a full spectrum of learning rules, and lead to the demonstration of the advantages of spiking computation over rate-coded computation.

II. Existence of Network Configuration

A. Definitions and notation

Assume that we have N neurons in a network. The network is desired to produce a given spatio-temporal firing pattern $\mathscr{P} = \{t_{ip} \,|\, 0 < t_{ip} \le T, i = 1,\cdots,N, p = 1,\cdots,k_i\}$ (see Fig. 1) of time duration T, in succession to a given initial pattern[1] $\mathscr{P}' = \{t_{ip} \,|\, t_{ip} \le 0, i = 1,\cdots,N, p \le 0\}$. In pattern $\mathscr{P}$ each neuron has k_i spikes ($i = 1,2,\cdots,N$).

Here we only consider leaky integrate-and-fire (LIF) neurons with fixed connection (synaptic) delays. The behavior of the membrane potential V_i of the ith LIF neuron, with a resting potential being zero, is given by the following differential equation [8]:

$$\frac{dV_i}{dt} = -\lambda V_i + I_i(t) \tag{1}$$

where $\lambda > 0$ denotes leakage coefficient and $I_i(t) \in \mathbb{R}$ denotes input current to the ith neuron. When V_i first crosses the threshold $\theta > 0$ from below, the neuron fires and V_i is reset to the resting potential 0.

Assume that the ith neuron fires at time t_{ip}. The ith neuron receives input spikes from the other $N-1$ neurons and a constant input I_i. Then the incremental potential of the ith neuron at time $t_{i(p-1)} \le t < t_{ip}$ is given as the following:

$$I_i(t) = I_i + \sum_{j=1, j\neq i}^{N} \sum_{t_{i(p-1)} - d_{ji} < t_{jq} \le t - d_{ji}} w_{ji}\,\delta(t - (t_{jq} + d_{ji})) \tag{2}$$

where t_{jq} is the qth spike of input neuron j, $w_{ji} \in \mathbb{R}$ is the connection weight from the source neuron j to neuron i, δ is Kronecker's delta function, and $d_{ji} > 0$ is the connection delay from neuron j to neuron i.[2] Inserting eq. (2) to eq. (1) yields

$$\frac{dV_i}{dt} = -\lambda V_i + I_i + \sum_{j=1, j\neq i}^{N} \sum_{t_{i(p-1)} - d_{ji} < t_{jq} < t - d_{ji}} w_{ji}\,\delta\big(t - (t_{jq} + d_{ji})\big) \tag{3}$$

The problem we are dealing with here is to find input strength I_i and weights w_{ji} $(i, j = 1,\cdots,N)$ of a network, so that the network generates the desired spatio-temporal pattern $\mathscr{P}$ commencing at time $t = 0$, given the connection delays d_{ji} and the initial condition (the initial pattern $\mathscr{P}'$). Note that only a part of the initial pattern is sufficient to determine the initial condition of the network, i.e. $\{t_{ip}\}$ $(-k_i' + 1 \le p \le 0)$, where $k_i' \ge 1$ is the number of spikes required for initial condition, which satisfies $t_{i(-k_i')} < \min_j(t_{j0} - d_{ji}) \le t_{i(-k_i'+1)}$.

B. Breaking down the problem

[1] Neurons in the network may have unknown external input before the last firing in the initial pattern, $t < t_{i0}$, but not after that.

[2] Although this notation cannot handle the cases of multiple synapses from neuron j to neuron i with different delays, it is straightforward to apply the argument to such cases.

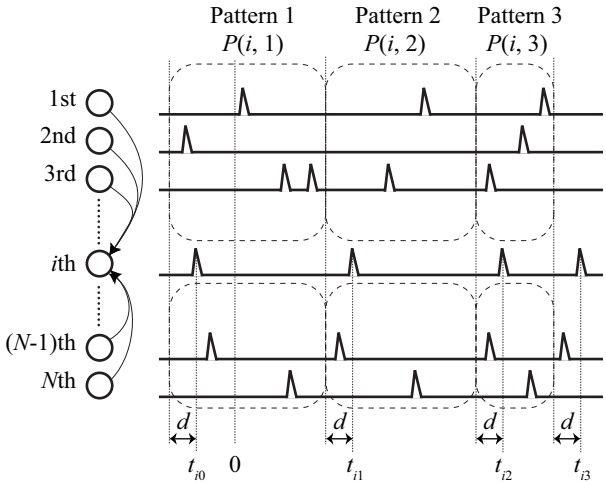


Fig. 1. Input patterns to the ith neuron. All synaptic connections have the same delay d. The input sequence is divided according to spikes of the desired output pattern of the ith neuron. Note that each neuron has different divisions.

To describe the required condition of the network, we break down the spatio-temporal pattern $\mathscr{P}$ into *input patterns*, each of which is related to one spike output. An input pattern $P(i,p)$ related to the firing t_{ip} of ith neuron is defined as

$$\begin{aligned} P(i,p) = \{&\{t_{1q}+d_{1i} : t_{i(p-1)} < t_{1q}+d_{1i} \le t_{ip}, 1 \neq i\}, \\ &\{t_{2q}+d_{2i} : t_{i(p-1)} < t_{2q}+d_{2i} \le t_{ip}, 2 \neq i\}, \\ &\vdots \\ &\{t_{Nq}+d_{Ni} : t_{i(p-1)} < t_{Nq}+d_{Ni} \le t_{ip}, N \neq i\}\}, \end{aligned} \tag{4}$$

which is the set of timing of all spike arrivals that cause depolarization or hyperpolarization to the ith neuron in the range $t_{i(p-1)} < t \le t_{ip}$ (see for example, Fig. 1). We say the ith neuron *responds* to an input pattern $P(i,p)$, remembering that the neuron does not fire during $t_{i(p-1)} < t < t_{ip}$, fires at t_{ip}, resets at time $t_{i(p-1)}$ and receives an input pattern $P(i,p)$ together with an input constant I_i.

Since the following lemma holds, we can write down the required condition of a network for each input pattern. Moreover, since the the parameters of ith neuron's input (I_i and w_{ji}) are independent of those of the other neurons', we can solve the problem one neuron at a time, instead of solving the whole network at once.

Lemma 1: Let NET be a deterministic N-neuron network having an initial condition at $t = 0$ given by the initial pattern $\mathscr{P}'$. NET reproduces the pattern $\mathscr{P}$ in succession to $\mathscr{P}'$ if and only if (a) every ith neuron ($i = 1,\ldots,N$) in the NET responds to every input pattern $P(i,p)$ ($p = 1,\ldots,k_i$) and (b) every ith neuron produces no spikes within the range $(t_{ik_i}, T]$.

This lemma is proved in Appendix.

In the rest of this article, we fix the neuron index i and pursue a way to find input strength I_i and weights w_{ji} ($j = 1,\cdots,N$) so that the ith neuron responds to the input patterns $P(i,p)$ ($p = 1,\ldots,k_i$).

C. Conditions

The ith neuron responds correctly to the pth input pattern (i.e. the neuron does not fire during $t_{i(p-1)} < t < t_{ip}$ and fires at t_{ip}) only if the membrane potential V_i satisfies the following two conditions:

$$V_i(t_{ip}) \geq \theta \qquad \text{(reaching condition)} \tag{5}$$

$$V_i(t) < \theta \quad (t_{i(p-1)} < t < t_{ip}) \qquad \text{(first-crossing condition)} \tag{6}$$

Both conditions are related to the potential at a designated time. However, assuming that the network obeys the designated pattern $\mathscr{P}$, we have the exact information of all input spikes to the ith neuron as well as reset timing of the neurons. Thus, we can write down $V_i(t)$, the potential of the ith neuron, as a function of time t with patameters I_i, w_{ji}, d_{ji} and $\mathscr{P}$. The ith neuron can respond to the given pattern only if this function $V_i(t)$ is consistent with the above two conditions. In this section, we show that the conditions can be represented in a set of linear equations and inequalities on I_i and w_{ji}.

By solving eq. (3), we can see that $V_i(t)$ is a linear summation of $V_i^C(t)$, the contribution from the constant current input, and $V_i^S(t)$, synaptic inputs from other neurons.

$$V_i(t) = V_i^C(t) + V_i^S(t) \tag{7}$$

The contribution of the constant input is

$$V_i^C(t) = I_i \cdot c_i(t) \quad , \tag{8}$$

where

$$c_i(t) = \frac{1}{\lambda}\left[1 - \exp\left(-\lambda\left(t - t_{i(p-1)}\right)\right)\right] \tag{9}$$

$V_i^C(t)$ increases from zero to I_i/λ provided that $I_i > 0$.

The contribution of the input from other $N-1$ neurons to the ith neurons is

$$V_i^S(t) = \sum_{j=1, j\neq i}^{N} w_{ji} \cdot s_{ji}(t) \quad , \tag{10}$$

where

$$s_{ji}(t) = \left[\sum_{t_{jq} \in P(i,p), t_{jq} \leq t} \exp\left(-\lambda\left(t - (t_{jq} + d_{ji})\right)\right)\right] \tag{11}$$

Thus, when we fix t, we can regard $V_i(t)$ as a linear function of I_i and w_{ji} $(j = 1, \cdots, N)$, their coefficients are given by $c_i(t)$ and $s_{ji}(t)$.

In the following, we investigate conditions (5) and (6).

1) Reaching condition, without spike arrivals synchronized with firings: Either a spike arrival synchronized to the firing time t_{ip} or current input can push the potential up to the threshold and satisfy (5). However, since we have the information of the pattern $\mathscr{P}$ and synaptic delay d_{ji}, we can tell whether the neuron i has the spike arrival synchronized at t_{ip} or not. In fact, when synaptic delays are determined independent of the pattern $\mathscr{P}$ (e.g. taken from some random distribution), the proportion of firings that accompanies synchronized spike arrival can be negligibly small. Moreover, in case that there is no such synchronized spike arrival, we can rewrite the condition (5) into an easily solvable equation.

Lemma 2: Consider the pth input pattern $P(i,p)$ which does not contain any spike arrival synchronized with post-synaptic firing, i.e., $t_{jq} \neq t_{ip} - d_{ji}$ for any q $(j = 1, \cdots, N)$. Then the following two conditins are necessary and sufficient for the reaching condition (5):

$$V_i(t_{ip}) = \theta \tag{12}$$

$$I_i > \lambda \cdot \theta \tag{13}$$

Note that eq. (13) means that the constant current acting alone would cause oscillation of the ith neuron.

Equation (12) becomes with Eqs. (8) and (10):

$$\theta = I_i c_i(t_{ip}) + \sum_j w_{ji} s_{ji}(t_{ip}) \quad . \tag{14}$$

This equation holds for all input patterns without spike arrival synchronized with post-synaptic firing. Let $P_{\mathrm{ASYNC}} = \{p_1, \ldots, p_{m_i}\} \subset \{1, \ldots, k_i\}$ be the set of indices of input patterns for ith neuron whose firing accompanies no synchronized post-synaptic firing, where m_i is the number of such patterns $(0 \leq m_i \leq k_i)$. Then eq. (14) gives m_i equations for variables I_i and w_{ji}. In matrix representation,

$$\begin{pmatrix} c_i(t_{ip_1}) & s_{1i}(t_{ip_1}) & \cdots & s_{Ni}(t_{ip_1}) \\ c_i(t_{ip_2}) & s_{1i}(t_{ip_2}) & \cdots & s_{Ni}(t_{ip_2}) \\ \vdots & \vdots & \cdots & \vdots \\ c_i(t_{ip_{m_i}}) & s_{1i}(t_{ip_{m_i}}) & \cdots & s_{Ni}(t_{ip_{m_i}}) \end{pmatrix} \begin{pmatrix} I_i \\ w_{1i} \\ w_{2i} \\ \vdots \\ w_{Ni} \end{pmatrix} = \begin{pmatrix} \theta \\ \theta \\ \vdots \\ \theta \end{pmatrix} \quad . \tag{15}$$

The equation above has a dimension of $m_i \times (N+1)$. Note that every element in the leftmost matrix is a constant. We call eq. (15) *the ith neuron equation* since all connections from the other neurons to the ith neuron and the current input I_i are fully determined by eq. (15).

The existence of a solution of ith neuron equation (eq. (15)) for all neurons $i = 1, \cdots, N$ is a necessary condition for the configuration of a spiking network that generates the desired pattern $\mathscr{P}$.

Thus this equation tells us that, for any given input/output pattern, we have to solve eq. (15) first to check whether a configuration of a spiking neural network is possible. In a scenario where the solution is unique, we can assess whether there is such a spiking neuronal network or not. If there is no solution, we see that there is no such a spiking neuronal network to generate the pattern P (see Example 1 below). If there is more than one solution, we have to find a way to choose one out of many (possibly infinite) solutions (see Example 1 and Section IV).

Example 1 Assume that there is no spike arrival synchronized with post-synaptic firing and that we have the following neuron equation (eq. (15)) for the 1st neuron:

$$\begin{pmatrix} c_i(t_{i1}) & 0 & \cdots & 0 & \cdots & 0 \\ c_i(t_{i2}) & s_{Ni}(t_{i2}) & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ c_i(t_{ik_i}) & 0 & \cdots & s_{ji}(t_{ik_i}) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & 0 & \ddots \end{pmatrix} \begin{pmatrix} I_i \\ w_{1i} \\ w_{2i} \\ \vdots \\ w_{Ni} \end{pmatrix} = \begin{pmatrix} \theta \\ \theta \\ \vdots \\ \theta \end{pmatrix} \quad , \tag{16}$$

which means that inside each interspike interval of the ith neuron, it receives exactly one spike from a different input neuron (but not the first neuron itself). When $k_i \geq N$, there is at most one unique solution (it is required that $t_{ik_i} = t_{i1}, k_i > N$). Hence we can easily design a desired spiking network to generate the required pattern P, provided that the first-crossing condition is satisfied (see the next subsection). When $k_i < N$, we have the freedom to choose weights $w_{ji}, j > k_i$ so that the first-crossing condition is fulfilled. Of course, the case of $k_i < N$ is more interesting and we will return to it in Section IV. Roughly speaking, when the number of neurons is greater than the number of patterns,

$$N > \max_{i=1,\cdots,N} k_i \tag{17}$$

we have the freedom to configure a spiking neuronal network to generate a desired spike pattern.

Equation (15) gives us only a necessary condition for producing $\mathscr{P}$; other conditions need to be satisfied to ensure the generation of the desired pattern, which is discussed below.

2) Reaching condition, when spike arrivals are synchronized with firings: In case an input pattern has a spike arrival synchronized with the post-synaptic firing, we cannot apply eq. (12), so we have to deal with the original inequality of the reaching condition

$$V_i(t_{ip}) \geq \theta \quad (p \notin P_{\mathrm{ASYNC}}, 1 \leq p \leq k_i) \quad . \tag{18}$$

Using eqs. (8) and (10), we can obtain an inequality for I_i and w_{ji}. Then, by combining with eq. (15), we can solve them to calculate the required configuration of the network.

3) First-crossing condition: To extract a set of inequalities for I_i and w_{ji} from the first-crossing condition (6), we can use the fact that eq. (13) guarantees the potential function $V_i(t)$ to be monotonically increasing in between any two incoming spike arrivals. For each input pattern $P(i,p)$, we can divide $V_i(t)$ into monotonically-increasing sections, and check that the sections have ends below the threshold. Since we know the timing of spike arrivals in desired patterns, it is enough to check eq. (6) only for some discrete times, immediate before any spike arrival.

Lemma 3: Suppose that ith neuron satisfies constraints (13), (15), and (18). The necessary and sufficient condition for holding the first-crossing condition (6) on the neuron is

$$\lim_{\varepsilon \to +0} V_i(t_{jq} + d_{ji} - \varepsilon) < \theta \qquad (t_{jq} \in P(i,p)) \quad . \tag{19}$$

Note also that we can obtain the constraint for proposition (b) in Lemma 1 by applying inequality (21) to arriving spikes in $(t_{ik_i}, T]$.

$$\lim_{\varepsilon \to +0} V_i(t_{jq} + d_{ji} - \varepsilon) < \theta \quad (j = 1, \ldots, N, j \neq i, t_{ik_i} < t_{jq} + d_{ji} \leq T) \tag{20}$$

If we merge the range of inequality (20) with those of inequalities (19) for all input patterns for ith neuron, it is equivalent to check the potential before all arriving spikes in the range $(t_{i0}, T]$.

$$\lim_{\varepsilon \to +0} V_i(t_{jq} + d_{ji} - \varepsilon) < \theta \quad (j = 1, \ldots, N, j \neq i, t_{i0} < t_{jq} + d_{ji} \leq T) \tag{21}$$

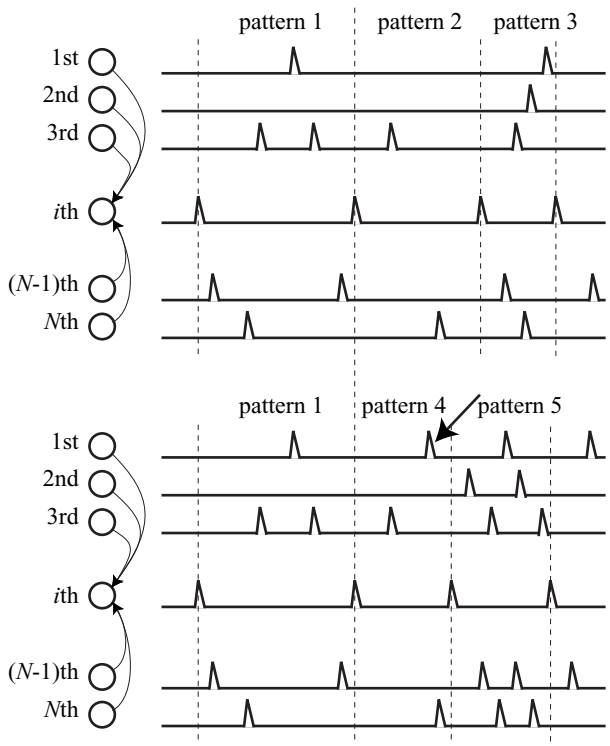


Fig. 2. Conditional switching. Pattern 2 and Pattern 4 are the same before a trigger input (the pointed spike). The network emits a totally different spike sequence based on the existence of the trigger.

By applying eqs. (8) and (10) to inequality (21), we can obtain a set of inequalities for I_i and w_{ji} that represents this first-crossing condition.

4) Network Configuration Theorem: By combining all conditions discussed in this section, we obtain the following theorem.

Theorem 4: Denote $\mathscr{P}$ as the desired pattern to be produced in succession to the initial pattern $\mathscr{P}'$, and T the duration of pattern $\mathscr{P}$. Let NET $= \langle (I_i), (w_{ji}), (d_{ji}) \rangle$ be a network with LIF neurons satisfying eq. (13), and having an initial condition at $t = 0$ given by the initial pattern $\mathscr{P}'$. NET generates the desired pattern $\mathscr{P}$ in succession to $\mathscr{P}'$ if and only if NET satisfies all constraints given by Eqs. (15), (18), and (21).

See Appendix for the proof of this theorem.

D. Extending patterns

So far, $P(i,p)$, the pth input pattern of the ith neuron, denotes a section in one continuous input sequence; but $P(i,p)$ can be used to denote other than that. In fact, for the ith neuron, every input pattern is independent. Thus we can use two or more patterns as a *conditional relation* of input and output (see Fig. 2).

Moreover the ith neuron can respond repeatedly to the same set of patterns. One of the simplest applications is a periodic sequence (see Fig. 3). In this case, the number of spikes in one period gives the number of patterns. If every neuron has a plentiful number of input neurons, an arbitrary long periodic pattern can be implemented (see Example 1).

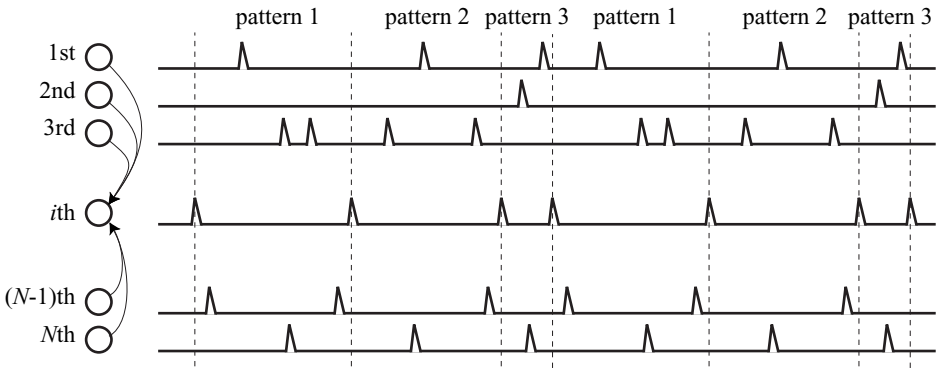


Fig. 3. Periodic sequence. Patterns are used repeatedly to construct a periodic sequence, which produces spikes more than the number of input neurons.

E. Multi-Layer Network

All results given above are independent of the network structure, i.e. they are established for recurrent networks. Here we have a close look at multi-layer networks.

Suppose that we have a multi-layered network and a neuron in the kth layer receives inputs from the $(k-1)$th layer in addition to other neurons in the kth layer. Then we can solve one layer at a time, assuming that the $(k-1)$th layer is correctly configured and send the desired spiking pattern to the kth layer. As we noted in Example 1, increasing the number of input connections will generally expand the solution space so that the first-crossing condition is more easily fulfilled.

In Section IV, we present numerical results for cases of one and two layers. For one layer cases, there is one input layer in addition to the output layer. The input layer has a few neurons which is directly driven by external input signals, and connected directly to the neurons in the output layer. There are no interactions between input neurons, but there are interactions between the output neurons. For three layer cases, one hidden layer is added.

III. Stability of the Pattern

In this section we discuss the issue of stability against *jitters*, i.e., fluctuation of temporal positions of the inputs. Here we confine ourselves to the stability within the given input pattern $P(i,p)$: We regard that the pattern is substantially changed when a spike in the pattern does not arrive or an extra spike is arrived. It is possible that different input patterns constitute the same spatio-temporal output pattern, in particular when spike arrivals are synchronized to the spike output; but we will not deal with such a case in this paper.

With noise (jitter), the actual firing time t_{ip} might be different from the desired firing time $\bar{t}_{ip}$. The difference $t_{ip}-\bar{t}_{ip}$, which is called the jitter of the firing, affects the subsequent activity of the network. The jitter can affect timing of subsequent spikes either by shift of reset timing or by shift of spike input timing. However, under certain conditions, small jitters would not change the order of subsequent spike arrivals and firings. This property enables us to investigate stability near the given input patterns.

In this section, we first inspect the condition under which small jitters do not change the order of spike

arrivals and firings. Then we discuss affection via the shift of reset timing. After that, we discuss affection via the shift of spike input timing. Finally, we discuss the stability of a whole pattern.

A. *Conditions for possibly stable output pattern*

The first step of stability analysis is to make sure that an infinitesimal jitter does not cause substantial changes to the behavior of the network. A substantial change, such as the excess or lack of a spike arrival in the given input pattern, may be caused by one of the following cases:

1) Violation of first-crossing condition: An infinitesimal jitter may perturb potential (details are discussed in the following sections). To ascertain that the perturbation does not violate inequality (21), we can impose a small finite margin $m > 0$ between the potential value and threshold. That is to say

$$V_i(t) \leq \theta - m \quad (t \neq t_{ip}, p = 1, \ldots, k_i) \quad . \tag{22}$$

Note that the reaching condition does not require special treatment like this. Although a small jitter may push the potential of ith neuron below the threshold at the firing time t_{ip}, the neuron will to fire shortly thanks to eq. (13). We will discuss this in Section III-C.

2) De-synchronization of synchronized spike arrivals: If a jitter is posed on a spike arrival that is synchronized with other spike arrivals, they get de-synchronized. In particular, if there is an excitatory spike input that is strong enough to induce firing but the firing is suppressed by synchronously arriving inhibitory spike input, jitter on either of them may put off the suppression and induce excess firing. Another case that needs special attention is two or more spike arrivals synchronized with the firing of the post-synaptic neuron; if the firing can be triggered without some of the synchronized spike arrivals, the rest of the spikes may arrive after the firing and change the potential that should be kept untouched after reset.

In either case, substantial change is caused by the firing of the post-synaptic neuron in absence of one of the synchronized spike arrivals. We can make a small modification on inequality (21) to suppress such firings:

$$V_i(t_{jq} + d_{ji}) - w_{ji} < \theta \qquad (t_{jq} \in P(i, p)), \tag{23}$$

where the left-hand side represents the potential of time t_{jq} without the spike arrival from jth neuron. If there are synchronized spike arrivals, this inequality is substantially more strict than the inequality (21), and is equivalent otherwise.

Combining the two considerations above, we obtain a new first-crossing condition that allows for a network that is resistant to infinitesimal spike jitters.

$$V_i(t_{jq} + d_{ji}) - w_{ji} \leq \theta - m \qquad (t_{jq} \in P(i, p)), \tag{24}$$

In the rest of this paper, we use this inequality as a first-crossing condition.

B. Jitter of neuron reset timing

A jitter of a spike on ith neuron affects subsequent spikes on the same neuron, because the jitter changes the reset timing. To see this relationship, we need to calculate the shift of the spike timing, dt_{ip}, in terms of the shift of the previous spike timing, $dt_{i(p-1)}$. To quantitatively describe it, we calculate $dV_i(\bar{t}_{ip})$, the change of potential at the expected firing time.

From Eqs. (7) and (9),

$$\begin{aligned} V_i(\bar{t}_{ip}) &= V_i^C(\bar{t}_{ip}) + V_i^S(\bar{t}_{ip}) \\ &= I_i \cdot \frac{1}{\lambda}[1 - \exp(-\lambda(\bar{t}_{ip} - t_{i(p-1)})] + V_i^S(\bar{t}_{ip}) \end{aligned} \tag{25}$$

Since V_i^S is independent of $t_{i(p-1)}$, we can derive

$$\frac{dV_i(\bar{t}_{ip})}{dt_{i(p-1)}} = -I_i \cdot \exp(-\lambda(\bar{t}_{ip} - t_{i(p-1)})) \tag{26}$$

On the other hand, $\frac{dt_{ip}}{dV_i(\bar{t}_{ip})} = -\left.\frac{dt_{ip}}{dV_i}\right|_{V_i=\theta}$, since $dV_i(\bar{t}_{ip})$ implies the change of function $V_i(t)$ itself but dV_i is just a gradient of the function. From eq. (3), we can see $\frac{dt_{ip}}{dV_i} = \frac{1}{I_i - \lambda V_i}$. Thus, the following equation holds.

$$\frac{dt_{ip}}{dV_i(\bar{t}_{ip})} = -\left.\frac{dt}{dV_i}\right|_{V_i=\theta} = -\frac{1}{I_i - \lambda\theta} \tag{27}$$

Combining this with eq. (26), we can derive the following relation.

$$\begin{aligned} \frac{dt_{ip}}{dt_{i(p-1)}} &= \frac{dt_{ip}}{dV_i(\bar{t}_{ip})} \cdot \frac{dV_i(\bar{t}_{ip})}{dt_{i(p-1)}} \\ &= \frac{\frac{I_i}{\lambda}}{\frac{I_i}{\lambda} - \theta} \exp(-\lambda(\bar{t}_{ip} - t_{i(p-1)})) \\ &= \frac{\frac{I_i}{\lambda} - V_i^C(\bar{t}_{ip})}{\frac{I_i}{\lambda} - \theta} \end{aligned} \tag{28}$$

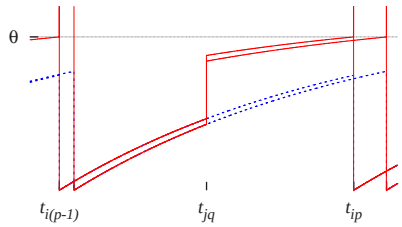
where V_i^C is the contribution of constant input to the potential, defined in eq. (9). From eq. (12), we can see $\theta = V_i(\bar{t}_{ip}) = V_i^C(\bar{t}_{ip}) + V_i^S(\bar{t}_{ip})$.

When a neuron has no spike input and fires only by a constant input, i.e. $V_i^S(\bar{t}_{ip}) = 0$, $\frac{dt_{ip}}{dt_{i(p-1)}}$ is in unity, which means that the amount of jitter is kept unchanged. When inhibitory spike inputs are dominant ($V_i^S(\bar{t}_{ip}) < 0$), then $\frac{dt_{ip}}{dt_{i(p-1)}}$ becomes less than 1, which means that the next jitter is smaller than the previous one, and $\bar{t}_{ip}$ can be a stable firing timing in case patterns are repeated. On the other hand, in a case the excitatory spike inputs are dominant ($V_i^S(\bar{t}_{ip}) > 0$), $\frac{dt_{ip}}{dt_{i(p-1)}} > 1$ holds and $\bar{t}_{ip}$ becomes an unstable firing timing.

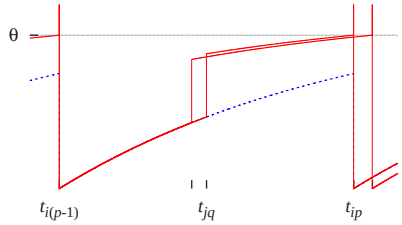
Figure 4(a) shows the case when there is only one excitatory input spike during the pattern. In this case $\frac{dt_{ip}}{dt_{i(p-1)}} \approx 2.25$ and the jitter of the next spike becomes larger than that of the previous one.

C. Jitter of spike input timing

A jitter of a spike on the ith neuron is also affected by jitter of spike arrival from another (jth) neuron. If a spike arrival is synchronized with post-synaptic firing in the pth input pattern and $V_i(\bar{t}_{ip})$ is substantially smaller than θ, the jitter of the last spike arrival is directly propagated to the output spike and the jitters of



(a) Delayed reset ($dt_{i(p-1)} > 0$)



(b) Advanced spike input ($dt_{jq} < 0$)

Fig. 4. Effect of jitters. In case of excitatory spike input, both delayed reset and advanced spike input cause delay of the next firing.

other spikes have only negligible effect. However, in other cases, the jitter of a spike arrival affects the jitter of the output spike in a non-trivial way. Here we calculate the factor $\frac{dt_{ip}}{dt_{jq}}$ to see the effect of small difference of input timing t_{jq} to the subsequent spike output timing t_{ip}. As in the previous section we calculate the factor via $dV_i(\bar{t}_{ip})$.

From Eqs. (7) and (11) (replacing some variables),

$$\begin{aligned} V_i(\bar{t}_{ip}) &= V_i^C(\bar{t}_{ip}) + V_i^S(\bar{t}_{ip}) \\ &= V_i^C(\bar{t}_{ip}) + \sum_{\substack{k=1 \\ k\neq i}}^{N} w_{ki} \left[\sum_{\substack{(t_{kr}+d_{ki})\in P(i,p) \\ t_{kr}+d_{ki}\leq t}} \exp(-\lambda(\bar{t}_{ip} - (t_{kr}+d_{ki}))) \right] \end{aligned} \tag{29}$$

Since all terms except $k = j$ and $r = q$ are independent of dt_{jq}, we can derive:

$$\frac{dV_i(\bar{t}_{ip})}{dt_{jq}} = w_{ji}\lambda(\exp(-\lambda(\bar{t}_{ip} - (t_{jq}+d_{ji}))) \tag{30}$$

Using eq. (27), we can calculate $\frac{dt_{ip}}{dt_{jq}}$ as in the following:

$$\begin{aligned} \frac{dt_{ip}}{dt_{jq}} &= \frac{dV_i(\bar{t}_{ip})}{dt_{jq}} \cdot \frac{dt_{ip}}{dV_i(\bar{t}_{ip})} \\ &= w_{ji}\lambda \cdot \exp(-\lambda(\bar{t}_{ip} - (t_{jq}+d_{ji}))) \cdot (-\frac{1}{I_i - \lambda\theta}) \qquad (31) \\ &= -\frac{1}{\frac{I_i}{\lambda} - \theta} w_{ji} \cdot \exp(-\lambda(\bar{t}_{ip} - (t_{jq}+d_{ji}))) \qquad (32) \end{aligned}$$

In other words, when a jitter is propagated from the jth neuron to the ith neuron, the amount of jitter is multiplied by $-\frac{1}{\frac{I_i}{\lambda}-\theta}$. This coefficient depends on the ith neuron and on $w_{ji}\lambda\exp(-\lambda(\bar{t}_{ip} - (t_{jq}+d_{ji})))$, the contribution of the jittered spike arrival to the potential of ith neuron at the timing of the subsequent spike.

Note that eq. (13) implies $\frac{I_i}{\lambda} > \theta$, and $s_{ji}(t) > 0$. Thus the sign of the jitter is negative if $w_{ji} > 0$.

Figure 4(b) shows the case when the input spike with large w has a jitter. In this case $\frac{dt_{ip}}{dt_{jq}} \approx -1.25$, and the jitter of the output spike becomes negative and slightly larger than that of the input spike. This shows nothing about the stability of the system; it depends on how the resulting jitter is propagated to another neurons.

Since the ith neuron has $N-1$ incoming synapses, the jitter of the ith neuron can be the sum of the jitters caused by the input.

$$\sum_i dt_{ip} = -\frac{1}{\frac{I_i}{\lambda} - \theta} \cdot \sum_j w_{ji} s_{ji}(\bar{t}_{ip}) dt_{jq} \tag{33}$$

If we represent the average of dt_{jq} as $\{dt\}$, eq. (33) can be approximated using eq. (10) as the following:

$$\begin{aligned} \sum_i dt_{ip} &= -\frac{1}{\frac{I_i}{\lambda} - \theta} \cdot \sum_j w_{ji} s_{ji}(\bar{t}_{ip}) \{dt\} \\ &= -\frac{1}{\frac{I_i}{\lambda} - \theta} \cdot V_i^S(\bar{t}_{ip}) \{dt\} \end{aligned} \tag{34}$$

From this equation, we can see two terms, which can be controlled to reduce propagation of jitters. One is $V_i^S(\bar{t}_{ip})$, which represents the contribution of input spikes in the potential at the firing times. Decreasing this term (e.g. remove some excitatory input or add some inhibitory input) makes the firing less sensitive to input jitters. Another is to increase $\frac{I_i}{\lambda}$, which represents the asymptotic potential caused by constant input. Increasing this term makes the ith neuron fire more regularly, and consequently, less sensitive to input jitters.

D. Stability of patterns

Finally, we study the stability of an autonomously repeated sequence. For simplicity, we focus on a simple sequence, in which every neuron fires once in a period. However, it is easy to extend the following discussion into more complex cases.

Let T be the length of one period in the repetition and $\bar{t}_i$ be desired firing time of the ith neuron in a period ($i = 1, \cdots, N$). We can re-number neurons so that $0 \le \bar{t}_1 < \bar{t}_2 < \cdots < \bar{t}_N < T$. Connection delay d_{ij} is less than the spike interval between the ith neuron and jth neuron, i.e. $d_{ij} < \bar{t}_j - \bar{t}_i$ $(i < j)$, $d_{ij} < \bar{t}_j - \bar{t}_i + T$ $(i > j)$.

Let δ_i be the jitter of the spike on the ith neuron. We would like to see δ_i', jitter of the spike in the next period, in terms of δ_i. From Eqs. (28) and (32), we can represent the relationship between δ_i and δ_i'.

$$\begin{aligned} \delta_i' = \frac{1}{\frac{I_i}{\lambda} - \theta} \Bigg[& \frac{I_i}{\lambda} \cdot \exp\big(-\lambda((T + \bar{t}_i) - \bar{t}_i)\big)\delta_i \\ & - \sum_{j=i+1}^{N} w_{ji} \exp\big(-\lambda((T + \bar{t}_i) - (\bar{t}_j + d_{ji}))\big)\delta_j \\ & - \sum_{j=1}^{i-1} w_{ji} \exp\big(-\lambda((T + \bar{t}_i) - (T + \bar{t}_j + d_{ji}))\big)\delta_j' \Bigg] \end{aligned}$$

We can convert this into a recurrence formula of $e^{\lambda \bar{t}_i}\delta_i$:

$$\begin{aligned} e^{\lambda \bar{t}_i}\delta_i' = e^{-\lambda T} \frac{1}{\frac{I_i}{\lambda} - \theta} \Bigg(& \frac{I_i}{\lambda}(e^{\lambda \bar{t}_i}\delta_i) - \sum_{j=i+1}^{N} w_{ji} e^{\lambda d_{ji}} (e^{\lambda \bar{t}_j}\delta_j) \\ & - \sum_{j=1}^{i-1} w_{ji} e^{\lambda(T + d_{ji})} (e^{\lambda \bar{t}_j}\delta_j') \Bigg) \end{aligned} \tag{35}$$

We introduce the following matrix notations:

$$\mathbf{v} = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_N \end{pmatrix} \qquad W_1 = \begin{pmatrix} 0 & e^{\lambda d_{12}} w_{12} & \cdots & e^{\lambda d_{1N}} w_{1N} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & e^{\lambda d_{N-1,N}} w_{N-1,N} \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

$$\mathbf{v}' = \begin{pmatrix} \delta_1' \\ \delta_2' \\ \vdots \\ \delta_N' \end{pmatrix} \qquad W_2 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ e^{\lambda d_{21}} w_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ e^{\lambda d_{N1}} w_{N1} & \cdots & e^{\lambda d_{N,N-1}} w_{N,N-1} & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} e^{\lambda \bar{t}_1} & 0 & \cdots & 0 \\ 0 & e^{\lambda \bar{t}_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & e^{\lambda \bar{t}_N} \end{pmatrix} \qquad I = \begin{pmatrix} \frac{I_1}{\lambda} & 0 & \cdots & 0 \\ 0 & \frac{I_2}{\lambda} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{I_N}{\lambda} \end{pmatrix}$$

Then we can rewrite eq. (35) as follows, where E is the N-dimensional identity matrix.

$$D\mathbf{v}' = e^{-\lambda T}(I - \theta E)^{-1}(ID\mathbf{v} - W_1 D\mathbf{v} - W_2 e^{\lambda T} D\mathbf{v}') \tag{36}$$

By simplifying this equation, we obtain the following:

$$\mathbf{v}' = e^{-\lambda T} D^{-1}(I - \theta E + W_2)^{-1}(I - W_1) D\mathbf{v} \tag{37}$$

Thus the change of the jitter after one period of the given sequence is represented by the matrix $M = e^{-\lambda T} D^{-1}(I - \theta E + W_2)^{-1}(I - W_1)D$. Using this matrix we can give the stability condition as follows.

$$\max |\text{eigenvalue}[M]| < 1 \tag{38}$$

When eq. (38) holds, $\lim_{n\to\infty} M^n = O$. This means that the jitter converges to zero after many repetitions of the sequence production, and hence, the system is asymptotically stable in repeating the given sequence.

Note that this stability depends on the order of firing. If the initial pattern has a large jitter and the firing order is changed, the above argument on stability does not hold anymore. This also means that a network can be stable in producing two or more sequences with different firing orders.

IV. Simulations

To illustrate the framework developed in the previous sections, we will show a few numerical simulations. For a given arbitrary spatio-temporal sequence, an appropriate configuration of a spiking neural network which produces the desired sequence is generated.

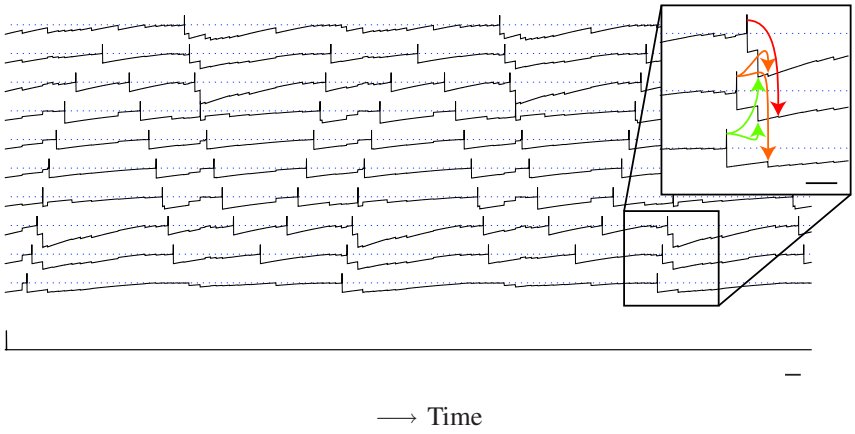


Fig. 5. Simulation of a network with a given repeated sequence. Input to the network is just one spike (thick line at the bottom) and constant current input. The bar at the bottom right corner indicates the length of the synaptic delay.

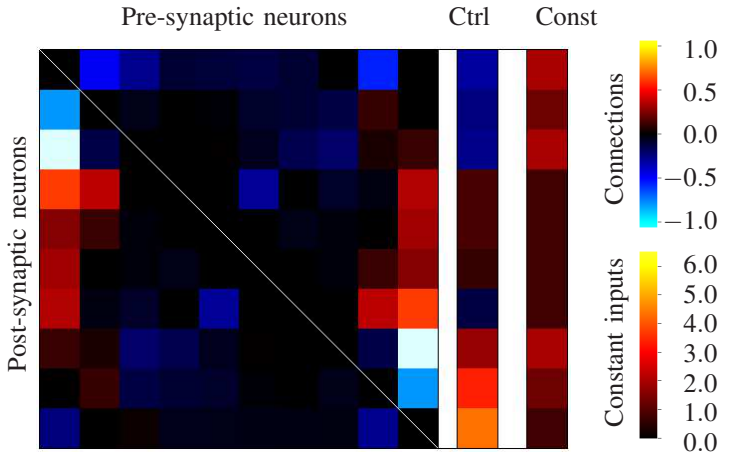


Fig. 6. Configuration of the network used in Fig. 5. "Ctrl" shows connection from control input (thick line in Fig. 5). "Const" shows constant current input, normalized by dividing by $\lambda \cdot \theta$.

A. Simulation Settings

Combining neuron equation (eq. (15)) with input current strength condition (eq. (13)), reaching condition on firing accompanying synchronized spike arrival (eq. (18)), and stable synchronized firing condition (eq. (24)), we have a set of linear inequalities for w_{ji}'s and I_i, which gives the range of possible configurations. Nevertheless, we usually have many (infinite) solutions for the configuration of spiking neural networks to generate the desired spikes, as discussed in Example 1. To pick one out of many solutions, we apply some optimization criteria here, i.e., to maximize the stability within all the possible configurations.

More specifically, here we use PCx [9], a linear-programming solver library, to calculate the connection weights and constant inputs for neurons. Since the condition of eq. (38) is too complex to be solved with linear-programming solver, we designed the cost function to be optimized based on the discussion in Sections III-B

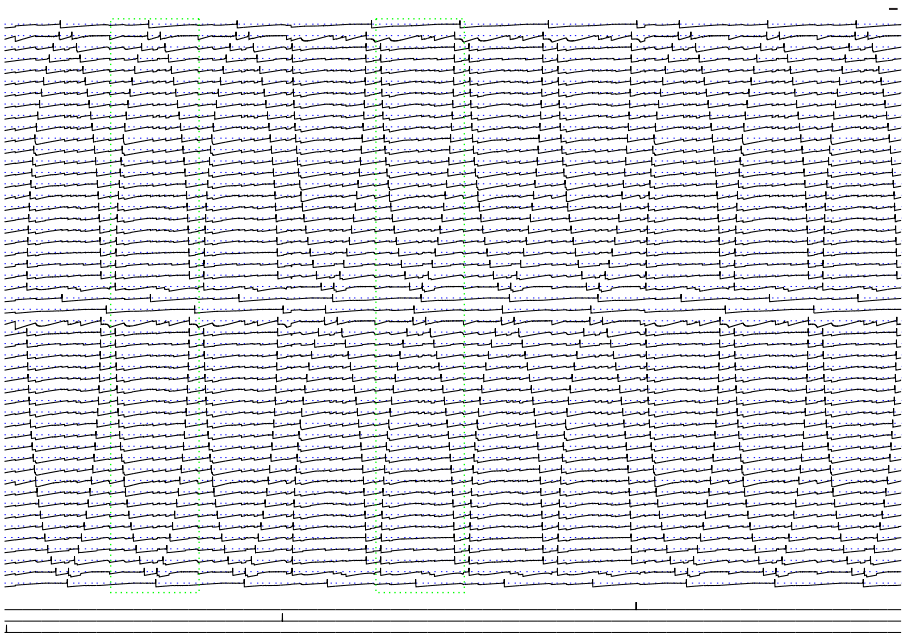


Fig. 7. Simulation of a network with switching sequences. Input to the network is three spikes (thick lines at the bottom) and constant current input. The bar at the top right corner indicates the length of the synaptic delay.

and III-C. The cost function for the ith neuron to be minimized is

$$z_i = \sum_{j=1}^{N} |w_{ji}| - C\frac{I_i}{\lambda} \tag{39}$$

where $C > 0$ is a constant to control the balance of optimization. In other words, the linear-programming solver prefers weak connection weights and a large I_i. With this cost function we can reduce the computational cost by dividing the original linear-programming problem into solving N smaller problems, one for each neuronal input. In all simulations, we set the delay d of all synaptic connections to $0.1l$, where l is the length of one periodic pattern, and the threshold margin m to 0.01θ. For simplicity, the initial condition we posed consists of (1) reset of potential without firing at time $t_{\text{begin}} < -d$, and (2) one trigger input at t_{trig} where $t_{\text{begin}} < t_{\text{trig}} < -d$. In other words, every neuron receives only one spike from the trigger input between the initial reset and the beginning of pattern output.

To simulate the configured network, we used PUNNETS simulation library [10] to perform the actual dynamics of the network. This library provides us with a very fine-grained temporal precision of simulating spiking neural networks using discrete-event (event-driven) simulation framework.

B. Simple Simulation

Figures 5 and 6 show a typical result of our simulations. In this simulation, we consider a network of ten neurons and configure them so that the network repeatedly produces the given spatio-temporal sequence, which looks like $\cap\cup\cap\cup\cdots$. As shown in Fig. 5, the given pattern is correctly recalled when the network receives the stimulus. The input neuron to the network (the thick line at the bottom of the figure) is feedforwardly

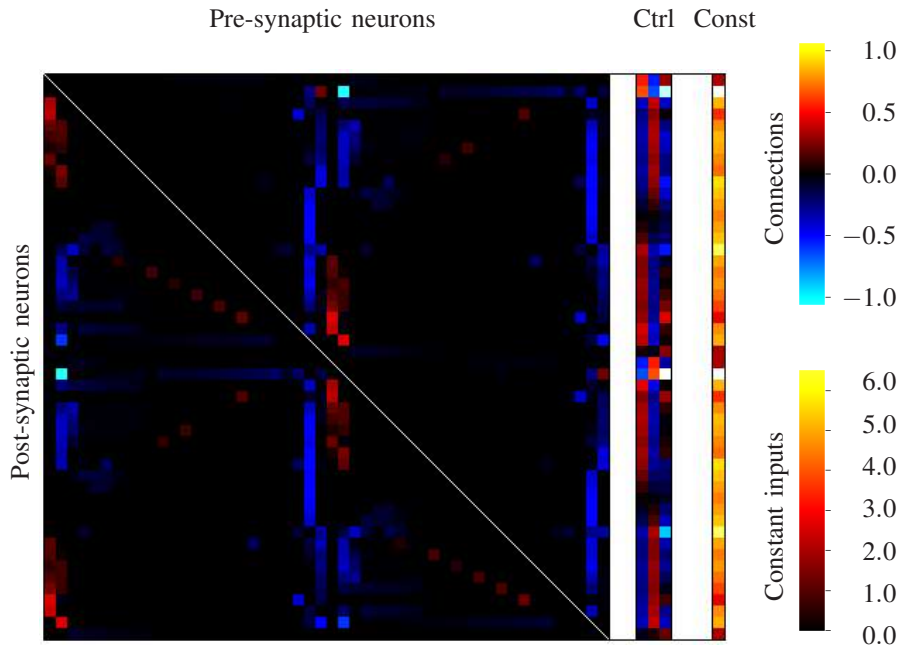


Fig. 8. Configuration of the network used in Fig. 7. "Ctrl" and "Const" are the same as in Fig. 6.

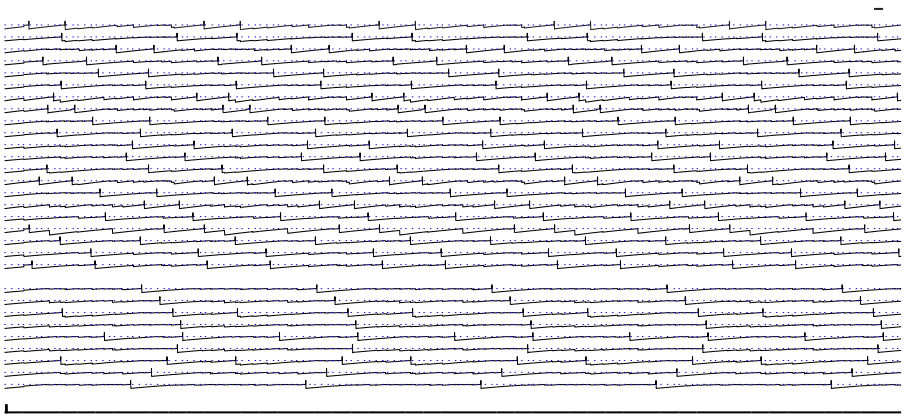


Fig. 9. Simulation of a network with multiple layers. The bar at the top right corner indicates the length of the synaptic delay.

connected to the ten output neurons and it sends one spike to initialize the network. The input spike triggers the subsequent network activity. The sequence of the desired pattern consists of a total of 44 patterns for 10 neurons.

Membrane potentials of each neuron are also depicted in Fig. 5. It clearly shows that when a neuron fires, it sends an EPSP (excitatory postsynaptic potential) or IPSP (inhibitory postsynaptic potential) to the neuron which it connects. It is interesting to note that usually the postsynaptic neuron does not fire simultaneously and the threshold crossing is mainly due to the constant input.

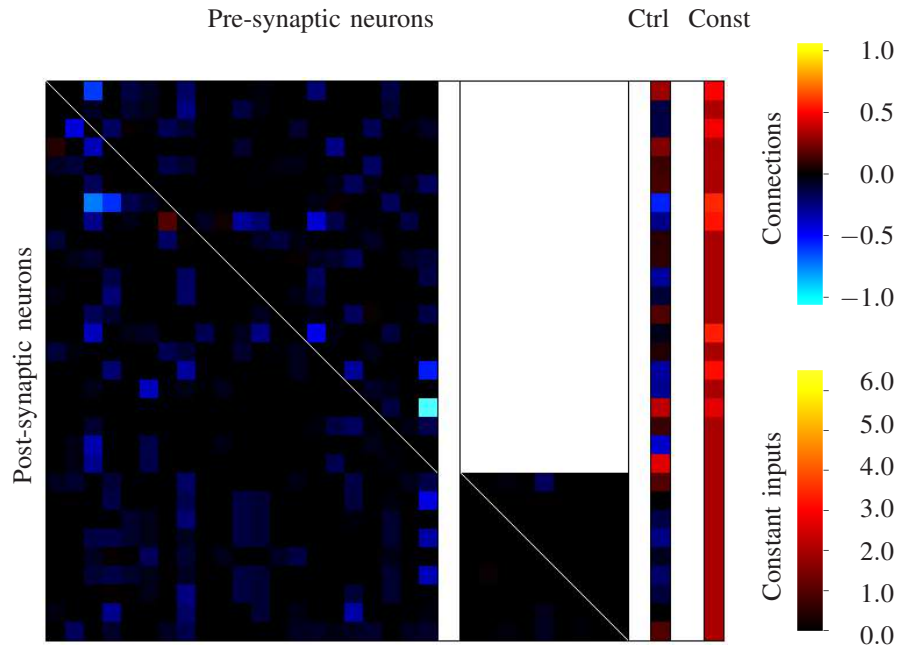


Fig. 10. Configuration of the network used in Fig. 9. "Ctrl" and "Const" are the same as in Fig. 6.

C. Large-scale Simulation

To further demonstrate the application of our approach, we consider more complex examples. Fig. 7 shows the result of another simulation. It is a network of 50 neurons and its activity switches between patterns. Three input neurons (thick lines at the bottom of the figure) are included in the network to initialize the activity. An O-like sequence is generated first, and the sequence repeats itself when it receives no further inputs. When the network receives another input spike then an X-like sequence is generated, and repeats itself if no further input is received. The sequence is then switched back to an O-like sequence again with the third input. All the sequences are broken down into a total of 338 patterns for the 50 neurons. The figure shows that the network correctly switches its output pattern according to trigger inputs of the input neurons.

In our simulation, the pattern OOO... is turned into XXX... with a single spike input, and turned back to OOO... with another single spike input. It is reminiscent of the creation of a working memory widely discussed in the literature.

D. Multi-layer Simulation

Fig. 9 shows the simulation of a two-layered network. The first layer is given a random repeating pattern, and the lateral connection within the layer is calculated by our formula. The second layer, which is given a repeating smiley-face pattern with the same period as the first pattern, has connections from the first layer as well as a lateral connection within the second layer. If the second layer had not had connections from the first layer, any configuration of this smiley face pattern we have found does not satisfy the stability condition. As a result, the network with such a configuration accumulates jitter quickly and its output pattern collapses

after a few repetitions. This example demonstrates that adding a layer of a random spiking pattern stabilizes production of a complex pattern.

V. Discussion

A. Feasibility

From our simulations, we found that some patterns were difficult to generate. It is likely to happen if (1) there are two close firings of a neuron with too few spike inputs between them, or (2) several patterns for the ith neuron share identical input neurons and input patterns. In case (1), solving the reaching condition gives an extremely large value of connection weights or constant input strength, and violates the first-crossing condition. Case (2) makes the ith neuron matrix linearly dependent, and causes eq. (15) to be infeasible. However, randomly given sequences with appropriate minimum distance between firings will not match either case, and as a result, will be almost always possible to be created.

Furthermore, the problem becomes more feasible by adding more neurons, as we have demonstrated in Section IV-D. Adding "supporting" neurons and assigning them some (possibly random) firing patterns associated with the desired firing sequences will break either infeasible case with some constant probability. By adding a plentiful amount of supporting neurons, we can make the possibility of infeasible cases to be as close to zero as we wish. Thus we can conclude *any* sequence can be produced in this way.

B. Stability

In some complex repeating sequences, we observed that the simulated network failed to produce the assigned sequence after several repetitions. We are able to analyze such instable patterns in terms of jitter increase; Note that the simulation error appears as jitters of spike timings because we used an event-driven simulation framework. Although the discussion of stability in Section III revealed the propagation of jitters, we cannot give good conditions to limit the coefficient of jitter propagation within the range (-1,1). It is one of our future topics to find a better way than eq. (39) to incorporate a stability condition (such as eq. (38)) into the solver.

C. Length of synaptic delay

In this paper, the only restriction on the synaptic delay is $\delta_{ij} > 0$. Indeed, we have no upper limits on the delay. For example, we have no problem to set δ_{ij} greater than the duration of a periodic sequence. All we need is to shift the input pattern with the amount of delay. Although introducing longer delay can increase the number of input patterns, which increases the number of constraints and hence makes the problem less feasible, we can add supporting neurons to make the problem more feasible as we discussed in Section V-A.

D. Sparse connectivity

As shown in the Section IV, the result of the connection calculated from the formula tends to be very sparse, a phenomenon widely discussed in the literature[11]. We observed that increasing the number of neurons improves sparseness. For example, in the 50-neuron example in Fig. 8, only 17.8% of the possible connections are connected. If we increase the number of neurons to 200, the connectivity decreases to 9.8%.

We can explain this sparseness based on our formula. To minimize the cost function (eq. (39)), weights of redundant connections tends to be zero, i.e. configured to be unconnected. eq. (16) tells us that, in general, the minimum number of non-zero post-synaptic connections to a neuron is the number of patterns of the neuron, e.g. the number of spikes in a period in case of simple repetition. Several extra connections are required to ensure first-crossing, and some more to further reduce the cost function. However, it seems that the number of these extra connections is affected only slightly by increasing the number of neurons.

E. Dominance of inhibitory connections

We also observed that more than half of the connections are negative (inhibitory synapses). In some easy configuration (like a random sequence), most of the connections become negative, as shown in Fig. 10. This is due to the formulation of the cost function (eq. (39)). With negative connections the solver can increase I_i to decrease the cost (i.e. make the network more stable). This is interesting because most lateral connections in the cerebral cortex are inhibitory [12].

F. Network model variants

In this paper we confine ourselves to a very simple model of spiking neural networks. However, our method is not restricted to such a simple model. In particular, we can see the ways to relax restrictions on step-like PSP (post-synaptic potential) .

No matter what form of post-synaptic potential is assumed, as long as the neuron model can be represented by Spike-Response Model [13], reaching conditions and first-crossing conditions can be described by a system of linear equations and inequalities, because the membrane potential is contributed by a linear sum of input spikes. There are basically no obstacles to extend our framework to include, for example, alpha-function-style PSP. However, additional effort would be required for discussing stability with these extensions.

G. Application of this framework

One of the simplest and most convincing applications of a configured spiking neural network is possibly in a control task, which requires an accurate replay of a given spatio-temporal pattern: for example the movement of an (artificial) arm. With a given task, we can first design a spatio-temporal pattern to control the movement and then apply our learning rule to configure a spiking network. This would possibly provide us with some evidence on the advantage of spiking neural networks over rate-coded ones.

Besides such a direct application, our formalizations could be informative in the study of neural learning and synaptic plasticity. Although we do not claim that a nervous system created a network exactly as our approach, the desired result of the learning is explicitly given by the Theorem 4. In particular, the ith neuron equation will enable us to study a new supervised learning rule, whose goal state is the given by the equation. For example, one may be able to develop a stochastic weight-update rule with a lowest energy state within the range of desired configuration.

VI. Related Work

A. Network dynamics

A lot of studies have been devoted to the dynamics of pulse-coupled oscillators, including spiking neural networks. Included are the stability of synchronous solution [14], [15], [16], and its coexistence with chaotic states [17], attracting behavior [18], [19], and the existence of multiple attractors [20], [21]. However, they rely on the given network configuration and discuss its output pattern. In this work, we made a different approach, which uses a given pattern and discuss required network configuration.

If we can convert the stable patterns in these studies into our notation of input pattern $P(i,p)$, the network satisfies the Network Configuration Theorem (Theorem 4). Problem is that the conversion is not always possible. Even if we have a conversion, in most cases, it is difficult to discuss the relation between network dynamics in those studies and that in our framework. Some of the reasons are listed below.

- Many studies concern systems without synaptic delay. We can treat such a system as a limited case of δ going to 0 only if the given spatio-temporal pattern includes no synchronous firing. However, if there are any synchronous firing in the given pattern, our framework cannot reproduce resetting behavior of neurons (e.g., none of the Models A to E in [18] becomes applicable).
- Most of these works are focusing on the dynamics about synchronous firing, or firings synchronous to spike arrivals. The ith neuron equation (eq. (15)), which is the core of our framework, is applicable only if the firing accompanies no synchronous spike arrivals, and cannot be discussed within the context of preceding studies. In addition, synchronous spikes may allow more than one input patterns to represent a spatio-temporal pattern with synchrony; in such a case, our stability analysis in Section III-D becomes too restrictive.
- In cases concerning integrate-and-fire oscillators without leakage (e.g., [18], [19], [17]), our stability analysis turns into a trivial result (by assuming $\lambda = 0$, eq. (28) becomes 1 and eq. (31) becomes 0).

Jin's work [22] is exceptional, because his analysis focuses on a fixed spiking sequence without synchronization on a network of leaky integrate-and-fire neurons. Indeed, his notion of "Pseudo Spiking Time" is similar to our discussion in Section III-D. Although there are small differences (Jin's network depends on global inhibition to avoid synchronous firing), it seems promising that his proofs can be extended to our framework.

B. Spatio-temporal pattern generation

One successful approach for spatio-temporal pattern generation is to use a recurrently connected network to produce a spatio-temporal pattern [23], [24]. However, with this approach, spikes can only have a discrete timing. Using a small time step will require an extremely large hidden layer, because this style of network requires a very long learning time for handling patterns with a long shared subpattern [25]. Our framework is advantageous because spikes can be at any timing in the spatio-temporal pattern.

C. Supervised Pattern Learning

One learning rule we should mention is the spike-timing-dependent plasticity (STDP), or temporally-asymmetric Hebbian learning rule [26], [27], [28]. We should note that STDP is an unsupervised learning rule that trains

a network to *predict* oncoming events [27]. In other words, STDP learns to compress a pattern in temporal dimension; it is different from a supervised learning for replaying a given spatio-temporal pattern without changing its length. We need another learning rule for applications that requires supervised learning of spatio-temporal patterns. Our work can be a good starting point for such research (see Section V-G).

VII. Conclusion

We derived a framework for finding the configuration parameter of a spiking neural network, such that the network produces the desired firing pattern with an appropriate initial condition. The framework was developed for the leaky-integrate-and-fire neurons with constant inputs. We calculated stability condition, and showed how to use a linear-programming routine to find a network configuration with an optimal stability. Simulation results to exhibit the behavior of the configured network to generate the complex patterns, including pattern switching, were presented.

Acknowledgments

We'd like to thank Prof. Kazuyuki Aihara for his time and patience in helping us. This research is partially supported by Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists.

Appendix

A. Proof of Lemma 1

Proof: In case that $\mathscr{P}$ contains no spike, the pattern is reproduced if, and only if, the NET produces no firing in the range $(0,T]$. This condition is equivalent to proposition (b).

Otherwise, Let t_0 be the time of the first spike in $\mathscr{P}$ and G be the set of neurons that has firing at time t_0 (i.e., $\forall i \in G \forall t_{jq} \in \mathscr{P}\,(t_0 = t_{i1} \leq t_{jq})$).

The part of the pattern $\mathscr{P}$ in the range $(0,t_0]$ is reproduced if, and only if, (1) neurons in G produces the first spike at time t_0, and (2) the other neurons produces no spike in the range $(0,t_0]$. Condition (1) holds because, from proposition (a), every ith neuron in G responds to $P(i,1)$. Condition (2) also holds for every ith neuron not in G (If $\mathscr{P}$ contains no spike on ith neuron, this is equivalent to proposition (b). Otherwise, proposition (a) says that the neuron responds to $P(i,1)$, so that the neuron does not fire in the range $(0,t_{i1})$, which is a superset of $(0,t_0]$). The rest of the pattern in the range $(t_0,T]$ can be proved by recursively applying the lemma, by shifting the pattern with the amount of $-t_0$ and transferring the first spike group $\{t_{i1} \,|\, i \in G\}$ from required pattern $\mathscr{P}$ to the initial pattern $\mathscr{P}'$. Since this operation decreases the number of spikes in the pattern $\mathscr{P}$, the whole lemma is proved from mathematical recursion. ■

B. Proof of Lemma 2

Proof: Sufficiency is trivial.

If we assume $V_i(t_{ip}) > \theta$, (6) is violated at the time just before t_{ip}, since there is no spike arrival at t_{ip}, $V_i(t)$ changes continuously in the neighbor of t_{ip}. Thus $V_i(t_{ip}) \leq \theta$ is necessary. Combining with eq. (5) yields eq.(12).

Moreover, since there is no spike arrival at t_{ip}, V_i cannot increase to θ if $I_i \leq \lambda \cdot \theta$. Thus inequality (13) is necessary. ■

C. Proof of Lemma 3

Proof: $V_i(t)$ $(t_{i(p-1)} < t \leq t_{ip})$ is divided into a finite number of continous sections by spike arrivals, $\{t_{jq} + d_{ji}\}$ $(t_{jq} \in P(i,p))$. Moreover, due to inequality (13), every continuous section in $V_i(t)$ is monotonically increasing. Thus, the condition (6) holds if and only if the latter end of every monotonically-increasing continuous section of $V_i(t)$ is below the threshold θ. Inequality (19) states this on all but the last section. If there is a spike arrival synchronized to the firing ($t_{jq} = t_{ip}$), the last section is also covered by inequality (19). In other cases, from eq. (12) (ensured by eq. (15)) and monotonic increase, we can derive $\lim_{\varepsilon \to +0} V_i(t_{ip} - \varepsilon) < \theta$. ■

D. Proof of Theorem 4

Proof: It is enough to show that the Eqs. (15), (18), and (21) are the necessary and sufficient condition of the propositions (a) and (b) in Lemma 1.

Proposition (a) is true if and only if reaching condition (5) and first-crossing condition (6) hold for all input pattern. As for reaching condition, Lemma 2 states that (5) is equivalent to (15) for patterns that contain no spike arrivals synchronized with post-synaptic firing. Inequality (18) states reaching condition (5) for the other patterns.

As for first-crossing condition, Lemma 3 states that inequality (19) is the necessary and sufficient condition, assuming that NET satisfies eqs. (15) and (18). Proposition (b) is equivalent to inequality (20), and the combination of inequalities (19) and (20) is equivalent to (21). ■

REFERENCES

[1] W. Singer, "Putative functions of temporal correlations in neocortical processing," in *Large-scale neuronal theories of the brain*, C. Koch and J. L. Davis, Eds. Cambridge, MA: MIT Press, 1994, pp. 201–238.

[2] H. Fujii, H. Ito, K. Aihara, N. Ichinose, and M. Tsukada, "Dynamical cell assembly hypothesis — theoretical possibility of spatio-temporal coding in the cortex," *Cognitive Science*, vol. 9, pp. 1303–1350, 1996.

[3] T. Makino, "A pulsed neural network for language understanding: Discrete-event simulation of a short-term memory mechanism and sentence understanding," Ph.D. Dissertation, Department of Information Science, Graduate School of Science, Tokyo University, Tokyo, Japan, 2001.

[4] K. W. Lee, H. Buxton, and J. F. Feng, "Cue-guided search: a computational model of selective attention,," *IEEE Transactions on Neural Networks*, 2004 (accepted).

[5] D. L. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Computation*, vol. 9, no. 4, pp. 805–836, 1997.

[6] W. Maass and C. M. Bishop, Eds., *Pulsed neural networks*. Cambridge, MA, USA: MIT Press, 1999.

[7] T. Nishikawa, Y. C. Lai, and F. C. Hoppensteadt, "Capacity of oscillatory associative-memory networks with error-free retrieval," *Physical Review Letters*, vol. 92, p. 108101, 2004.

[8] W. Gerstner, "Spiking neurons," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. Cambridge, MA: MIT Press, 1998, ch. 1, pp. 3–53.

[9] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright, "PCx: An interior-point code for linear programming," *Optimization Methods and Software*, vol. 11/12, pp. 397–430, 1999.

[10] T. Makino, "A discrete-event neural network simulator for general neuron models," *Neural Computing & Applications*, vol. 11, pp. 210–223, 2003.

[11] A. Destexhe and E. Marder, "Plasticity in single neuron and circuit computations," *Nature*, vol. 431, pp. 789–795, 2004.

[12] M. Abeles, *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge, UK: Cambridge University Press, 1991.

[13] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, MA: Cambridge University Press, 2002.

[14] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math*, vol. 50, pp. 1645–1662, 1990. [Online]. Available: http://tam.cornell.edu/SStrogatz_bio_oscillators_sync.pdf

[15] H. Torikai and T. Saito, "Synchronization phenomena in pulse-coupled networks driven by spike-train inputs," *IEEE Trans. Neural Networks*, vol. 15, no. 2, pp. 337–347, 2004.

[16] S. R. Campbell, D. Wang, and C. Jayaprakash, "Synchrony and desynchrony in integrate-and-fire oscillators," *Neural computation*, vol. 11, no. 7, pp. 1595–1619, 1999. [Online]. Available: http://openurl.ingenta.com/content?genre=article&issn=0899-7667&volume=11&issue=7&spage=1595&epage=1619

[17] M. Timme, F. Wolf, and T. Geisel, "Coexistence of regular and irregular dynamics in complex networks of pulse-coupled oscillators," *Phys. Rev. Lett.*, vol. 89, no. 25, p. 258701, 2002.

[18] J. J. Hopfield and A. V. M. Herz, "Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire neurons," *Proc. Natl. Acad. Sci. USA*, vol. 92, pp. 6655–6662, 1995. [Online]. Available: http://www.pnas.org/cgi/content/abstract/92/15/6655

[19] W. Gerstner, "Rapid phase locking in systems of pulse-coupled oscillators with delays," *Physical Review Letters*, vol. 76, no. 10, pp. 1755–1758, 1996.

[20] X. Guardiola and A. Dìaz-Guilera, "Pattern selection in a lattice of pulse-coupled oscillators," *Physical Review E*, vol. 60, pp. 3626–3632, 1999.

[21] M. I. Rabinovich, A. Volkovskii, P. Lecanda, R. Huerta, H. D. I. Abarbanel, and G. Laurent, "Dynamical encoding by networks of competing neuron groups: Winnerless competition," *Physical Review Letters*, vol. 87, no. 6, pp. 068 102/1–4, 2001.

[22] D. Z. Jin, "Fast convergence of spike sequences to periodic patterns in recurrent networks," *Physical Review Letters*, vol. 89, p. 208102, 2002.

[23] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum, 1986, pp. 531–546.

[24] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–221, 1990.

[25] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Computation*, vol. 1, pp. 372–381, 1989.

[26] L. F. Abbott and S. Song, "Temporally asymmetric Hebbian learning, spike timing and neural response variability," in *Advances in neural information processing systems*, M. S. Kearns, S. Solla, and D. Cohn, Eds. Cambridge, MA: MIT Press, 1999, vol. 11, pp. 69–75.

[27] R. P. N. Rao and T. J. Sejnowski, "Spike-timing-dependent hebbian plasticity as temporal difference learning," *Neural Computation*, vol. 13, pp. 2221–2237, 2001. [Online]. Available: http://openurl.ingenta.com/content?genre=article&issn=0899-7667&volume=13&spage=2221&epage=2237

[28] S. Song, "Hebbian learning and spiking-toming-dependent plasticity," in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed. London, Boca Raton: Chapman and Hall/CRC Press, 2004, pp. 305 – 340.