
ReSuMe - New Supervised Learning Method for Spiking Neural Networks

Filip Ponulak

Institute of Control and Information Engineering
Poznan University of Technology
Piotrowo 3a, Poznan, Poland
Filip.Ponulak@put.poznan.pl

Technical Report, Poznan, 2005

Available at: <http://dl.cie.put.poznan.pl/~fp/>

Abstract

In this report I introduce ReSuMe - a new supervised learning method for Spiking Neural Networks. The research on ReSuMe has been primarily motivated by the need of inventing an efficient learning method for control of movement for the physically disabled. However, thorough analysis of the ReSuMe method reveals its suitability not only to the task of movement control, but also to other real-life applications including modeling, identification and control of diverse non-stationary, nonlinear objects.

ReSuMe integrates the idea of learning windows, known from the spike-based Hebbian rules, with a novel concept of remote supervision. General overview of the method, the basic definitions, the network architecture and the details of the learning algorithm are presented. The properties of ReSuMe such as locality, computational simplicity and the online processing suitability are discussed. ReSuMe learning abilities are illustrated in a verification experiment.

1 Motivation

Control of posture and movement in the biological neural systems is a complex, nonlinear task. The neural mechanisms underlying control are still merely understood. The key role in that task is attributed to the plasticity and learning abilities of the Central Nervous System (CNS) [1].

The question on how the motor control system performs its tasks becomes pertinent, when facing the problem of CNS disorders causing the subjects' disability in grasping, standing or walking. In many cases the diminished functions can be augmented by the modern rehabilitation techniques, such as Functional Electrical Stimulation (FES) or Functional Neuromuscular Stimulation (FNS) [2, 3, 4, 5]. However, there is still a great challenge for engineers to design appropriate controllers meeting the requirements arising in FES and

FNS systems [2]. Such controllers are supposed to be robust and flexible. A special emphasis should be put on their good learning abilities and the adaptability to non-stationarities and nonlinearities of the human neuro-musculo-skeletal system. Thus, new, effective learning methods suitable for reconstructing of the neural signals (temporal sequences of spikes) are required.

Spiking Neural Networks (SNN) [6, 7] seem to be a very promising tool for such neuro-controllers synthesis.

SNN represent a special class of the Artificial Neural Networks, in which the neuron models communicate via spike trains. In SNN the timing of individual spikes plays crucial role [8]. Thus the relevant information is coded there in a similar manner as in the biological neural and neuro-muscular structures. It has been proved that the spiking neurons are computationally more powerful than perceptrons and sigmoidal gates [9, 10].

However, recent supervised learning methods in SNN are not suitable for the considered control tasks.

Most of the works in this area have focused on the gradient-following approach [11, 12, 13, 14, 15]. However, since explicit calculation of the gradient in SNN is unfeasible, usually special simplifications are postulated: in [14] it has been assumed that time scale is discrete, methods presented in [11, 12] or [13] require that the learning algorithms work in a range where all postsynaptic potentials are linear functions of time. In [15] it was proposed to evaluate the stochastic gradient. However, in this method the neuron firing was modeled by the Poisson process, which in the biological neurons holds only for the specific conditions.

All above simplifications constrain severely the use of the gradient-following learning methods.

In this report a definitely different approach is presented. A new learning technique, called ReSuMe (Remote Supervised Method), for the Spiking Neural Networks (SNN) is introduced.

Instead of computing gradient, ReSuMe takes advantage of the learning windows idea [6] integrated with a novel concept of remote supervision. It is shown experimentally that this approach has interesting learning properties desirable from the point of view of the considered control tasks.

The learning window has been introduced first for the Hebbian learning in the context of spike-based learning rules such as Spike-Timing Dependent Plasticity (STDP) [16, 17].

The original Hebbian rule and STDP are unsupervised learning techniques. For this reason they are not suitable for the applications that require an explicit goal definition. However, STDP rule has some important properties that would be desirable also for the supervised learning methods.

First of all STDP is a local rule. The local character of the learning mechanism enables the scalability of STDP. Since the synaptic weights are updated in the incremental manner, the method is suitable for the on-line processing. Another consequence of the STDP locality is the computational simplicity of STDP implementations.

For these reasons it is interesting to define a new learning method for SNN which inherits the advantages of the STDP approach while enabling supervision.

In the next chapter Remote Supervised Learning Method is introduced. It is argued that ReSuMe possesses the expected properties and for this reason is suitable for the real-life, real-time applications. The presentation of ReSuMe begins with the general overview of the method. Next, the basic definitions and the learning algorithm are introduced. Finally, the properties of ReSuMe are illustrated in a verification experiment.

2 ReSuMe Introduction

The goal of ReSuMe learning is to impose on a neural network the desired input-output properties, i.e. to produce the desired spike trains in response to the given input sequences.

The ReSuMe learning algorithm can be understood as follows. Consider a subset N^l of all neurons in a spiking neural network. N^l consists of neurons, which update their input synaptic connections according to ReSuMe method. Each neuron in N^l is excited simultaneously with a number of spike trains through multiple synapses. Assume that for each of these neurons an individually assigned single signal with the predetermined timing of spikes is defined - this is the reference (teacher) signal desired at the learning neuron's output.

The ReSuMe learning proceeds by balancing two opposite rules determined over every synapse terminating at the considered learning neurons (Fig.1). The rules have the forms similar to STDP and anti-STDP mechanisms, respectively [6]. The rules are expressed as functions, called learning windows, of the difference between: presynaptic and reference spikes times (the first rule) or pre- and postsynaptic spikes times (the second rule).

According to the first rule, an excitatory (inhibitory) synapse is facilitated (depressed) if it transmits a presynaptic spike directly before the reference spike time (Fig.1.B.1). According to the second rule an excitatory (inhibitory) synapse is depressed (facilitated) if a presynaptic spike arrives directly before a postsynaptic spike (Fig.1.B.2). The combination of these rules allows to obtain the desired timing of spikes at the learning neurons with a very high precision.

The learning rules are local. Thus ReSuMe is scalable and can be applied in the same form to the simple structures of neurons as well as to the large, complex networks.

In the presented learning method, neurons that deliver the desired signals (teacher neurons) are not directly connected to the learning neurons. However, they supervise the learning synapses, i.e. their activity determines the synaptic weights modification (Fig.1.A). For this reason the proposed learning approach is called a Remote Supervised Method (ReSuMe).

2.1 Basic Definitions

There are three types of neurons that take part in the learning process: input, learning and teacher neurons. A set of input neurons $N^{in} = (n_1^{in}, n_2^{in}, \dots)$ represents the neurons activating the learning synapses. A set of learning neurons $N^l = (n_1^l, n_2^l, \dots)$ consists of the neurons that receive signals from the learning synapses and are expected to produce the desired signals $S^d(t)$. The signals $S^d(t)$ are delivered to the network via set of teacher neurons: $N^d = (n_1^d, n_2^d, \dots)$.

Learning triple. For every learning neuron n_i^l we define a subset $N^{in}(i) \in N^{in}$ of input neurons activating the neuron n_i^l . We also specify a single teacher neuron n_j^d delivering the desired signals for the neuron n_i^l (we denote it by $n_j^d(i)$). Hence we can define a *learning triple* $L_i = (N^{in}(i), n_i^l, n_j^d(i))$ for each learning neuron i .

Spike train. We formally specify signals transmitted in Spiking Networks. Let $t_m^{(f)}$ be the firing times of a neuron m (where $f=1,2,\dots$ is a label of each individual spike emitted by the neuron). According to [6] a spike train of a neuron m is defined as a sequence of the firing times:

$$S_m(t) = \sum_f \delta(t - t_m^{(f)}), \quad (1)$$

where $\delta(x)$ is the impulse function ($\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ elsewhere).

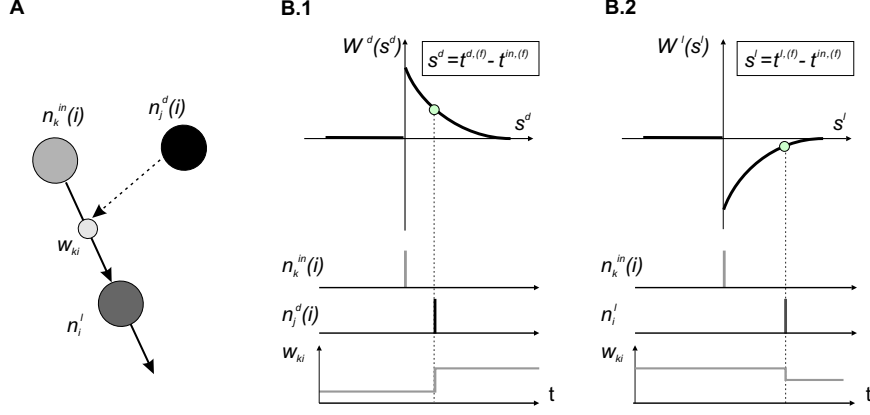


Figure 1: Two concepts underlying ReSuMe learning: **(A) Remote supervision.** In ReSuMe the synaptic efficacy w_{ki} , between any given presynaptic neuron $n_k^{in}(i)$ and a corresponding postsynaptic neuron n_i^l , depends not only on the correlation between the pre- and postsynaptic firing times, but also on the correlation of activities between $n_k^{in}(i)$ and a 'remote' teacher neuron $n_j^d(i)$. **(B) Learning windows.** Changes of the synaptic efficacy w_{ki} are triggered by teacher or postsynaptic action potentials at times $t^{d,(f)}$ and $t^{l,(f)}$, respectively. The amplitude of change is determined by the functions $W^d(t^{d,(f)} - t^{in,(f)})$ and $W^l(t^{l,(f)} - t^{in,(f)})$, called learning windows.

Learning windows. In ReSuMe, for any synaptic connection w_{ki} from a neuron n_k^{in} to n_i^l , the synaptic weight is updated at the firing times of n_i^l or $n_j^d(i)$. The amplitude of modification is determined by two functions $W^d(s^d)$ and $W^l(s^l)$ where $s^d = t^{d,(f)} - t^{in,(f)}$ and $s^l = t^{l,(f)} - t^{in,(f)}$. The functions $W^d(s^d)$ and $W^l(s^l)$ are *learning windows* [6]. In ReSuMe these functions are exponential (Fig.1.B). Such shape of the learning window is based on the experimental physiological observations [18].

Performance measure. In order to quantitatively estimate the performance of learning, we define three measures of the quality of the desired signal approximation. The first measure, called performance index, expresses the global distance between the trained and the desired signals. The second measure defines the precision of $S^d(t)$ approximation during the ReSuMe training. The third measure gives an amount of the time-shift error between individual spikes in $S^d(t)$ and the corresponding $S_i^l(t)$ signal.

The process of learning during a single presentation of a pair of patterns (input/desired output) is called a learning session. For the m -th learning session we define a *performance index* $P(m)$ as:

$$P(m) = \int |L(S^d(t)) - L(S^l(t))| dt \quad (2)$$

where for any spike train $S(t)$ defined by (1), $L(S(t))$ denotes a lowpass filtering:

$$L(S(t)) = \sum_f \exp\left(\frac{-t + t^{(f)}}{\tau}\right) \cdot H(t - t^{(f)}) \quad (3)$$

$H(x) = 0$ for $x < 0$, $H(x) = 1$ elsewhere; τ is a filter time constant.

The performance index can be viewed as a measure of a distance between the considered spike sequences. It takes high values for the poor quality of fit while it decreases to zero for $S^l(t) = S^d(t)$.

Denote by $\hat{S}^l(t)$ an output signal obtained after the ReSuMe training. We say that $\hat{S}^l(t)$ approximates $S^d(t)$ with the *precision* r if the following conditions are satisfied:

1. the number of spikes in $\hat{S}^l(t)$ and in $S^d(t)$ is equal.
2. for each $t^{d,(f)} \in S^d(t)$ there exists one and only one $t^{l,(f)} \in \hat{S}^l(t)$ such that $|t^{d,(f)} - t^{l,(f)}| \leq r$,

The precision parameter r is a real positive value. It is reasonable to expect the value of r less than a half of a minimal inter-spike interval in $S^d(t)$ in order to avoid an incorrect assignment of the same spike in $\hat{S}^l(t)$ to more than one spike in $S^d(t)$.

Finally, for the signal $\hat{S}^l(t)$ we introduce a *spike-shift error* $e(t)$ which gives the measure of a distance between the corresponding spikes in $\hat{S}^l(t)$ and $S^d(t)$. The spike-shift error $e(t)$ is defined as a vector of pairs $(t^{d,(f)}, \Delta t^{(f)})$ where $t^{d,(f)}$ is a time of the f -th spike in the desired spike train $S^d(t)$ and $\Delta t^{(f)}$ is a time difference between $t^{d,(f)}$ and the time of the nearest spike in $\hat{S}^l(t)$ (an example of $e(t)$ is illustrated in Fig.3.F).

2.2 Learning algorithm

Learning in ReSuMe is performed by modifying the synaptic efficacies w_{ki} of every synaptic connection between an input $n_k^{in}(i) \in N^{in}(i)$ and a learning neuron n_i^l . The modification algorithm is applied according to the following equation:

$$\begin{aligned} \frac{d}{dt} w_{ki}(t) = & S^d(t) \left[a^d + \int_0^\infty W^d(s^d) S^{in}(t - s^d) ds^d \right] \\ & + S^l(t) \left[a^l + \int_0^\infty W^l(s^l) S^{in}(t - s^l) ds^l \right], \end{aligned} \quad (4)$$

The (real-valued) constants a^d and a^l determine amplitudes of, so called, non-Hebbian processes of weight modifications. For the excitatory synapses $a^d > 0$ and $a^l < 0$, whereas for the inhibitory synapses $a^d < 0$ and $a^l > 0$.

The integral functions in (4) represent the Hebbian contributions to the weight change: the first one depending on the correlation between the presynaptic and teacher activities ($S^{in}(t), S^d(t)$) and the second one depending on the correlation between the pre- and post-synaptic signals ($S^{in}(t), S^l(t)$). To express these two correlation factors, respectively, we use learning windows $W^d(s^d)$ and $W^l(s^l)$ introduced in the previous section (where again $s^d = t_j^{d,(f)} - t_k^{in,(f)}$ and $s^l = t_i^{l,(f)} - t_k^{in,(f)}$). Here we propose a simple choice for the learning windows, inspired by [18] and optimized to ensure fast convergence of the ReSuMe algorithm:

$$W^d(s^d) = \begin{cases} +A^d \cdot \exp\left(\frac{-s^d}{\tau^d}\right) & \text{if } s^d > 0, \\ 0 & \text{if } s^d \leq 0, \end{cases} \quad (5)$$

$$W^l(s^l) = \begin{cases} -A^l \cdot \exp\left(\frac{-s^l}{\tau^l}\right) & \text{if } s^l > 0, \\ 0 & \text{if } s^l \leq 0, \end{cases} \quad (6)$$

where A^d, A^l and τ^d, τ^l are some constants. For the excitatory synapses A^d and A^l take the positive real values, whereas for the inhibitory synapses A^d, A^l are negative. In both cases the learning time constants τ^d, τ^l are real and positive. The learning windows defined by (5) and (6) are illustrated in Fig.1.B.

If we now set: $a^l = -a^d$, $A^l = A^d$ and $\tau^l = \tau^d$, then Eq.(4) takes the following form:

$$\frac{d}{dt}w_{ki}(t) = [S^d(t) - S^l(t)] \left[a^d + \int_0^\infty W^d(s^d) S^{in}(t - s^d) ds^d \right], \quad (7)$$

Equation (7) reveals that the synaptic modifications are essentially driven by the difference $[S^d(t) - S^l(t)]$ between the desired and generated signals.

Assign by D the whole time domain in which the synaptic plasticity of $w_{ki}(t)$ is observed. At any time $t \in D$ the term $[a^d + \int_0^\infty W^d(s^d) S^{in}(t - s^d) ds^d] \neq 0$. On the other hand $[S^d(t) - S^l(t)] = 0$ if and only if $t^{l,(f)} = t^{d,(f)} = t$. Thus, the synaptic efficacy in (7) remains unmodified for the whole time domain D , that is $\forall_{t \in D} dw_{ki}(t)/dt = 0$, if and only if $\forall_{t \in D} S^l(t) = S^d(t)$. This conclusion states that eq.(7) reaches the fixed point only for the learning signal $S^l(t)$ equal to the desired signal $S^d(t)$. It can be shown that under certain conditions this fixed point is a global, positive attractor in the weight space.

2.3 Network Architecture

For any learning triple $L_i = (N^{in}(i), n_i^l, n_j^d(i))$ there are two aspects of learning that must be taken into account in order to reconstruct the desired spike train $S_j^d(t)$ at the learning neuron n_i^l :

1. ensuring optimal synaptic weights at the input (presynaptic) connections to n_i^l ,
2. ensuring that the set of signals driving n_i^l and the trained output signal $\hat{S}_i^l(t)$ are in such a relation that every desired spike $t_i^{l,(f)} \in \hat{S}_i^l(t)$ is directly preceded by at least one excitatory presynaptic spike¹ entering n_i^l .

The first condition is expected to be satisfied by the learning procedure itself while the latter one must be satisfied by designing an appropriate network architecture.

Assign the set of signals driving the neural network by $S^{in}(t)$ and the signals that input a given learning neuron n_i^l by $\hat{S}_i^{in}(t)$ (cf. Fig.2). We require a subnetwork performing a specific, unique mapping from $S^{in}(t)$ onto $\hat{S}_i^{in}(t)$ such that for any arbitrary chosen pairs of $S^{in}(t)$ and $\hat{S}_i^l(t)$ and for all $n_i^l \in N^l$ condition 2 is satisfied with respect to $\hat{S}_i^{in}(t)$.

Such mapping can be performed in different network architectures. Recently ReSuMe has been applied and used (successfully) for training in sparse and in fully connected recurrent networks.

As an example, we present here an implementation of ReSuMe in the Liquid State Machine (LSM) architecture proposed by Maass et al. (see e.g. [19]). The Liquid State Machine consists of a large, fixed "reservoir" network - the neural microcircuit (NMC) from which the desired output is obtained by training the suitable output connection weights. The properties of LSM and NMC are described in details in [19].

In the implementation of ReSuMe method the original LSM approach has been modified. The modified architecture consists of a set of input neurons N^{in} , the NMC structure, a set of learning neurons N^l and a corresponding set of teacher neurons N^d (Fig.2). NMC receives signal $S^{in}(t)$ from N^{in} and transforms it into a vector of signals $\hat{S}_i^{in}(t)$ which is presented to the adequate learning neurons $n_i^l \in N^l$. The teacher neurons N^d are not

¹This condition is necessary, since the special cases such as bursting neurons or huge currents injection are not considered here.

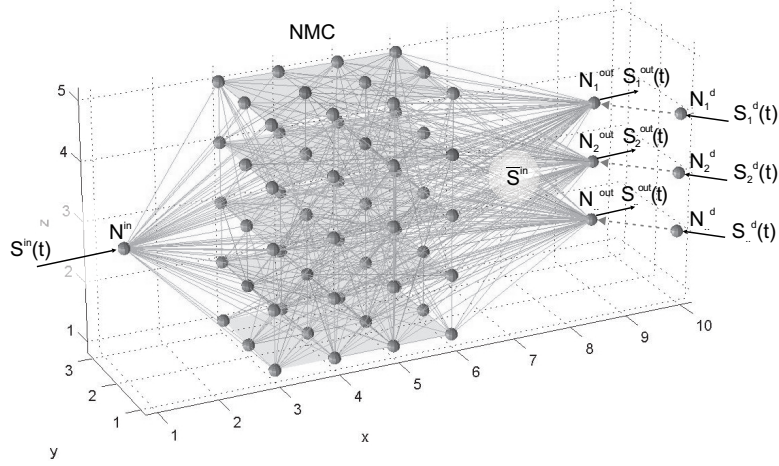


Figure 2: ReSuMe implemented in the modified Liquid State Machine Architecture. The network consists of a set of input neurons N^{in} , the NMC structure, a set of learning neurons N^l and a corresponding set of teacher neurons N^d . NMC receives signal $S^{in}(t)$ from N^{in} and transforms it into a vector of signals $\hat{S}_i^{in}(t)$ which is presented to the learning neurons N^l . The learning neurons are supervised by the corresponding teacher neurons N^d that deliver individually assigned reference patterns to each $n^l \in N^l$.

directly connected with any other structure, however the correlation of activity of the particular $n_i^d \in N^d$ with $\hat{S}_i^{in}(t)$ determines the modification of the synaptic efficacy of the connections between NMC and n_i^l (as it was described in the previous section).

3 Experimental verification

In this section I present the results of an experiment that confirms an ability of ReSuMe to efficiently train the network to produce the desired output pattern $S^d(t)$ in response to the given, specified input spike sequence $S^{in}(t)$.

The experiment was performed in CSIM: A neural Circuit SIMulator [20]. Both $S^{in}(t)$ and $S^d(t)$ signals were generated randomly over a time interval of 400 ms. The input sequence $S^{in}(t)$ (Fig.3.A) was presented to NMC through the input unit that projected on all neurons of NMC. NMC consisted of 800 Leaky-Integrate-and-Fire (LIF) neurons [6] with the connections generated randomly according to the procedure described in [19]. NMC responded to the input sequence with the unique state trace \hat{S}^{in} (Fig.3.B). This trace was a driving input to a single readout neuron.

The readout was trained over 100 learning sessions to produce the desired spike train $S^d(t)$ (Fig.3.D). At every learning session m the same pairs of $S^{in}(t)$, $S^d(t)$ signals were presented to the network, the learning neuron signal $S^l(t)$ was recorded and a performance index $P(m)$ was calculated. An output signal $S^l(t)$ at the initial state of learning and the trained output $\hat{S}^l(t)$ are depicted in Fig.3.C and 3.E respectively.

During the training the quality of $S^d(t)$ approximation improved significantly from $P(1) = 114.25$ to $P(100) = 3.87$ (Fig.3.G). Starting from the 75-th learning session all spikes of $S^d(t)$ were correctly recalled at the learning neuron and the spike times of the produced output were only slightly shifted in relation to the desired times. After the training an

average spike-shift-error $e(t)$ was 0.65 ms and the maximal error did not exceed 2 ms (compare to 8 ms of the minimal interval between the neighboring spikes in the desired signal) (Fig.3.F).

4 Conclusions

In this report a new supervised learning method for SNN has been presented. The method called ReSuMe introduced a novel approach to the supervised learning techniques by integrating the idea of learning-windows with the novel concept of remote supervision.

The experiment presented here confirmed that ReSuMe can efficiently learn the desired temporal sequences of spikes and that the learning process converges quickly.

Further studies on ReSuMe indicate that the method enables to learn multiple patterns of spikes. This can be performed sequentially or in parallel, by assigning different patterns to the particular outputs of the trained network. Since the method is based on the correlation of spike times, it is expected that ReSuMe should work properly not only for the LIF models, but also for other, more complex, models (e.g. Hodgkin-Huxley models) of spiking neurons. This was also experimentally verified [21].

The performance of the learning process in ReSuMe depends among others on the learning window parameters and the appropriate matching of the network size to the complexity of the learning task (determined by the number of spikes, the length and number of the learning patterns, etc.). The learning performance is determined also by an ability of the reservoir-network (in the LSM architecture - of the neural microcircuit) to transform any given input signal into the network state such that the state projection onto a learning unit covers uniformly, with the desired density of spikes, the time domain required for the given learning tasks.

At the moment it cannot be uniquely determined whether the proposed mechanism of "remote supervision" is biologically plausible. However, there are many physiological evidences for, so called, *heterosynaptic plasticity* [18], [22], [23], in which the induction of synaptic modifications at one set of synapses can be accompanied by changes at some neighboring synapses that did not experience the induction activity. This phenomenon is similar to the concept of "remote supervision" and can contribute to determining the biological processes potentially underlying this concept.

General features of Spiking Neural Networks and the particular properties of ReSuMe suggest the suitability of ReSuMe for the eventual applications to the real-world, real-time tasks.

Acknowledgements

The work was partially supported by the State Committee for Scientific Research, project 1445/T11/2004/27.

References

- [1] E. R. Kandel, T. M. J. Schwartz, and T. M. Jessel, editors. *Principles of neural sciences*. Elsevier, New York, 1991.
- [2] D. Popovic and T. Sinkjaer. *Control of Movement for the Physically Disabled*. Springer-Verlag, 2000.
- [3] J. Abbas and J. Gillette. Using Electrical Stimulation to Control Posture. *IEEE Control Systems Magazine Special Issue*, 21(4):80–90, 2001.
- [4] U. Stanic and R. Davis. General Consideration in the Clinical Application of Electrical Stimulation. International FES Society web page: <http://www.ifess.org>.

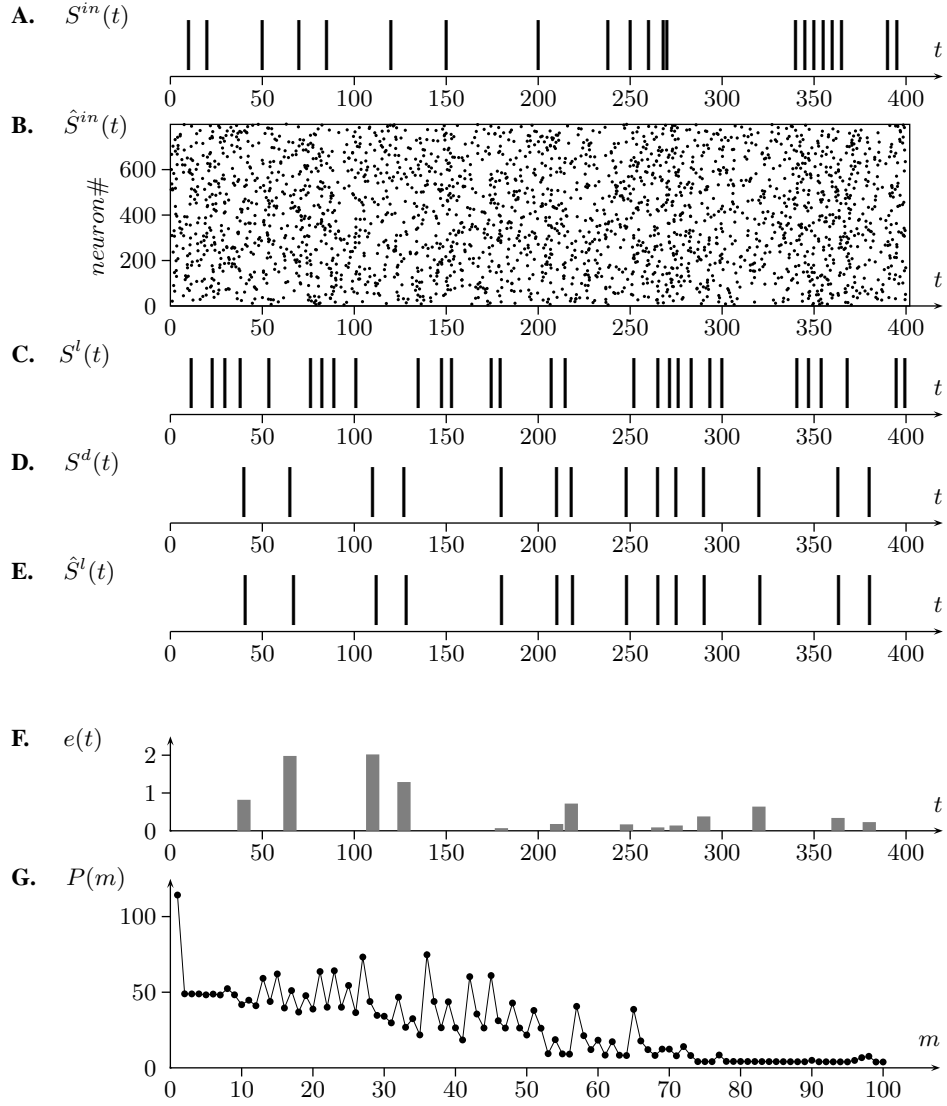


Figure 3: The network of integrate-and-fire neuron models was trained to produce a desired spike train $S^d(t)$ in response to an input $S^{in}(t)$. Both signals of length 400ms were generated randomly. The network consisted of a single input, single output, a teacher neuron and the NMC structure with 800 units. The system was trained over 100 learning sessions. (A) Input spike train $S^{in}(t)$. (B) NMC state trace $\hat{S}^{in}(t)$ resulting from $S^{in}(t)$. The consecutive rows of points indicate the spike trains generated by the NMC neurons (labeled here with numbers). (C) Output $S^l(t)$ before the training. (D) Spike train $S^d(t)$ desired at the output. (E) Output $\hat{S}^l(t)$ after the training. (F) Spike-shift error $e(t)$ between $S^d(t)$ and $\hat{S}^l(t)$. Error (given in [ms]) is plotted as a function of time. (G) Performance index $P(m)$ in the consecutive learning sessions m .

- [5] Juliusz Huber, Przemyslaw Lisinski, Andrzej Kasinski, Monika Kaczmarek, Piotr Kaczmarek, Przemyslaw Mazurkiewicz, Ponulak Filip, and Wojtysiak M. Therapeutic Effects of Spinal Cord and Peripheral Nerve Stimulation in Patients with Movement Disorders. *Artificial Organs*, 28(8):766, August 2004. (Abstract).
- [6] Wulfram Gerstner and Werner Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge, 2002.
- [7] Wolfgang Maass and Christopher Bishop, editors. *Pulsed Neural Networks*. The MIT Press, Cambridge, 1999.
- [8] Rufin VanRullen, Rudy Guyonneau, and Simon J. Thorpe. Spike times make sense. *TRENDS in Neurosciences*, 28(1):1–4, 2005.
- [9] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. In *Proc. of the 7th Australian Conference on Neural Networks, in Canberra, Australia*, pages 1–10, 1996.
- [10] Wolfgang Maass. Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 211–217. MIT Press (Cambridge), 1997.
- [11] Wolfgang Maass and Thomas Natschlaeger. Networks of spiking neurons can emulate arbitrary hopfield networks in temporal coding. *Network: Computation in Neural Systems*, 8:355–372, 1997.
- [12] Berthold Ruf and Michael Schmitt. Learning temporally encoded patterns in networks of spiking neurons. *Neural Processing Letters*, 5(1):9–18, 1997.
- [13] S.M. Bohte, J.N. Kok, and H. La Potr’e. SpikeProp: Backpropagation for Networks of Spiking Neurons. In *European Symposium on Artificial Neural Networks, ESANN*, pages 775–777, 2000.
- [14] David Barber. Learning in spiking neural assemblies. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 149–156. MIT Press, Cambridge MA, 2003.
- [15] Xiaohui Xie and Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review*, 69(041909):1–10, 2004.
- [16] Werner Kistler. Spike-timing dependent synaptic plasticity: a phenomenological framework. *Biol.Cybern.*, 87:416–427, 2002.
- [17] Wulfram Gerstner and Werner Kistler. Mathematical formulations of Hebbian learning. *Biol.Cybern.*, 87:404–415, 2002.
- [18] Guo-Qiang Bi. Spatiotemporal specificity of synaptic plasticity: cellular rules and mechanisms. *Biological Cybernetics*, 87:319–332, 2002.
- [19] Wolfgang Maass, Thomas Natschlaeger, and Henry Markram. Computational models for generic cortical microcircuits. In J. Feng, editor, *Computational Neuroscience: A Comprehensive Approach*, pages 575–605. Chapman and Hall/CRC, Boca Raton, 2004. Chapter 18.
- [20] Thomas Natschlaeger, Henry Markram, and Wolfgang Maass. Computer models and analysis tools for neural microcircuits. In R Koetter, editor, *Neuroscience Databases. A Practical Guide*, pages 123–138. Kluwer Academic Publishers (Boston), 2003. Chapter 9.
- [21] Andrzej Kasinski and Filip Ponulak. Experimental Demonstration of Learning Properties of a New Supervised Learning Method for the Spiking Neural Networks. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Biological Inspirations*, volume 3696 of *Lecture Notes in Computer Science*, pages 145–153, 2005.
- [22] Hui zhong W. Tao, Li I. Zhang, Guo qiang Bi, and Mu ming Poo. Selective presynaptic propagation of long-term potentiation in defined neural networks. *The Journal of Neuroscience*, 20(9):3233–3243, May 2000.
- [23] Tobias Bonhoeffer, Volker Staiger, and Ad Aertsen. Synaptic plasticity in rat hippocampal slice cultures: local ‘hebbian’ conjunction of pre- and postsynaptic stimulation leads to distributed synaptic enhancement. *Proceedings of National Academy of Science USA*, 86:8113–8117, October 1989.