34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Synaptic Dynamics: Linear Model and Adaptation Algorithm

Ali Yousefi, Alireza A. Dibazar, and Theodore W. Berger, *Fellow, IEEE*

Abstract— **Linear model for synapse temporal dynamics and learning algorithm for synaptic adaptation in spiking neural networks are presented. The proposed linear model substantially simplifies analysis and training of spiking neural networks, meanwhile accurately models facilitation and depression dynamics in synapse. The learning rule is biologically plausible and is capable of simultaneously adjusting both of LTP and STP parameters of individual synapses in a network. To prove efficiency of the system, a small size spiking neural network is trained for generating different spike and bursting patterns of cortical neurons. The simulation results revealed that the linear model of synaptic dynamics along with the proposed STDP based learning algorithm can provide a practical tool for simulating and training very large scale spiking neural circuitry comprising of significant number of synapses and neurons.**

I. Introduction

Synaptic dynamics is the computational engine for processing natural time-varying stimuli in brain [1, 2]. Processing of the time-varying stimuli is a difficult problem, and it is yet an unsolved problem for artificial models of brain function. Synaptic efficacy changes on a short time scale by several hundred percent in dependence of the past pre-synaptic activity, and it is the balance of facilitation and depression that determines the synaptic temporal dynamics and thereby forming the basis of neural computation [3]. Mathematical models of synaptic dynamics formulate facilitation and depression mechanism of neurotransmitter release replicating synapse temporal dynamics [3, 4, 5, 6, 7]. Facilitation-depression models (FD) represent long term plasticity (LTP) and short term plasticity (STP) of a synapse in a unified model; in which the LTP and STP are mutually dependent characteristics of synapse regulating its efficacy and temporal dynamics. Despite of biological plausibility of FD models, non-linearity and stochasticity of the proposed models make systematic analysis of synapse temporal dynamics cumbersome.

Learning process infers the proper balance of facilitation, depression and neurotransmitter quanta in each synapse of an artificial neural network to achieve a desired spatio-temporal computation [3, 8]. Learning in dynamic synapses neural network (DSNN) challenges different difficulties. Spike generation is not a differentiable function of the neuron membrane activity restricting gradient descent approaches in synaptic adjustment. Also, defining an objective function to characterize similarity between spike trains is not straightforward [9], and different similarity measures change learning process drastically. Nonlinearity and time-variant of spiking neural network is the other obstacle of the learning process. In fact, there is a lack of well-established learning algorithms for DSNN, meanwhile none of the proposed learning algorithms addresses STP adaptation. Tempetron [10] and ReSuMe [11] learning process only handle LTP adaptation in single layer of spiking neuron. SpikeProp [12] and Adeli et al. [13] are applicable to spiking neural network where there is only one spike per neuron per processing period. In fact, these learning algorithms - and their counterparts- are unable to exploit fine spatio-temporal computation of DSNN.

Through this paper, two breakthrough steps in building an artificial neural model capable to solve the problem of processing time-varying stimuli are introduced; 1) a linear synapse model and 2) a biologically plausible synaptic adaptation algorithm. The novel model presents a balance of computational simplicity and biological plausibility, and it is applicable in DSNN with different time scales. The network provides a spatio-temporal computation unit suitable for neural simulation and pattern recognition tasks.

In the first section of the paper, it is shown that the FD model- proposed by Markram et al.[5]- can be linearized, and yet presents an accurate model of the synapse dynamics. Also, a systematic sensitivity analysis for synapse temporal dynamics in response to change in the synapse internal parameters and impinging spike pattern is developed. The linear model plus sensitivity analysis is the substantial tool for analysis and learning of DSNN.

In the second section of the paper, a biological plausible learning algorithm is defined [14]. The learning algorithm is motivated by a supervised learning in which synaptic parameters are adjusted to increase similarity between the network output and a desired spike train. The learning algorithm is a reward-based STDP learning rule [2] and is applicable for single layer recurrent DSNN.

Through the third section of the paper, a specific application of the model in neural modeling is addressed. It is shown that synaptic parameters of a network consisting only two synaptic connections can be adjusted to reproduce different spiking and bursting patterns of the rat cortical neurons reported by Izhikevich [15].The novel neural model can be applied for a larger number of synaptic connections providing a computation tools for spatio-temporal processing tasks.

II. Linear Model of Synapse Dynamics

FD models for synapse dynamics [5, 6, 7] have the following general structure:

* This research was supported by the awards from ONR (Award No: N00014-10-1-0685) and the Navy through a CPP grant (Award No: N00014-09-C-0209).

Ali Yousefi. is PhD candidate in Neural Dynamics Laboratory , University of Southern California; email: ayousefi@usc.edu

Alireza A. Dibazar is Research Assistant Professor and Co-Director of Neural Dynamics Laboratory, University of Southern Californian.

Theodore W. Berger is David Packard Professor in Biomedical Eng. Department of University of Southern California.

$$F_{n+1} = A_{11,n} * F_n + B_{1,n} \quad (1.a)$$
$$N_{n+1} = A_{22,n} * N_n + A_{23,n} * F_n * N_n + B_{2,n} \quad (1.b)$$
$$G_{n+1} = A_{42,n} * N_n + A_{43,n} * F_n * N_n + A_{44,n} * G_n \quad (1.c)$$
$$K_{n+1} = A_{52,n} * N_n + A_{53,n} * F_n * N_n + A_{54,n} * K_n \quad (1.d)$$
$$V_n = G_n - K_n \quad (1.e)$$

Variable F_n determines the facilitation process at time index n and N_n represents release ready vesicle quanta. Variable V_n denotes post synaptic membrane potential (PSP) and is determined by two auxiliary variables G_n and K_n. $A_{-,n}$ and $B_{-,n}$ coefficients are function of synapse internal parameters plus time of impinging action potentials (APs). Appendix A represents a detailed definition for each of $A_{-,n}$ and $B_{-,n}$ coefficients. In synapse model, both F_n and N_n are normalized values ranging from zero to one; and their values are controlled by synapse parameters and history of APs.

Synapse dynamics is AP-driven process. In response to a AP, facilitation factor increases rapidly and decays with a time constant about 100 milliseconds. Instantaneously, the increased facilitation determines quanta of vesicle release, which is defined by $F_{n+1} * N_n$. Release ready vesicles in synaptic cleft are depleted by $F_{n+1} * N_n$ and recovered by a time constant of around 200 milliseconds. The PSP- either excitatory or inhibitory - amplitude is proportionate to released vesicle per AP and is modeled with an α-function [16]. Temporal dynamics of synapse is emerged by interplay of facilitation increase plus facilitation and vesicle recovery time constants.

Synpase model written in equation (1) is a non-linear time-variant system. Synapse dynamics between spike time intervals is a linear time-invariant system, and it is AP occurrence injects non-linearity and time-variant to synapse model [2, 14]. Non-linearity in synapse model appears in $F_n * N_n$ term; and it becomes linear if $F_n * N_n$ is replaced by a slack variable. The slack variable $R_n = F_n * N_n$ is defined, which denotes minimum vesicle release for possible AP at time $n+1$. To build the synapse linear state space model, R_{n+1} needs to be defined by a linear combination of F_n, N_n and R_n. Variable $R_{n+1} = F_{n+1} * N_{n+1}$ can be rewritten as linear function of F_n, N_n, R_n plus a new non-linear term - ${F_n}^2 * N_n$. In the approximate linear model, the ${F_n}^2 * N_n$ term is dropped. This is a practical approximate as the ${F_n}^2 * N_n$ is smaller than R_n- because $F_n \leq 1$- and has less contribution than other synapse variables in building R_{n+1}. Appendix A represents the state matrix and input vector for the linearized synapse model. It is possible to derive more accurate linear state space model by changing ${F_n}^2 * N_n$ to another slack variable and dropping higher order terms. Equation (2) represents synapse linear state space model:

$$X_{n+1} = \left(A_s + 1_{ap}(n) * A_t\right) * X_n + \left(B_s + 1_{ap}(n) * B_t\right) \quad (2.a)$$
$$V_n = [0 \quad 0 \quad 0 \quad 1 \quad -1] * X_n = C * X_n \quad (2.b)$$

$X_n = [F_n \quad N_n \quad R_n \quad G_n \quad K_n]'$ is the model state vector and V_n corresponds to post synaptic membrane potential. Elements of A_s, A_t, B_s and B_t are constant function of synapse internal parameters, and $1_{ap}(n)$ is one at AP time. The linear model provides a compact tool for temporal and steady state analysis of the synapse. Two following subsections analyze synapse temporal dynamics as a function of synapse internal parameters and AP variability. Figure (1.b) shows PSP for different facilitation factors in response to a periodic APs.

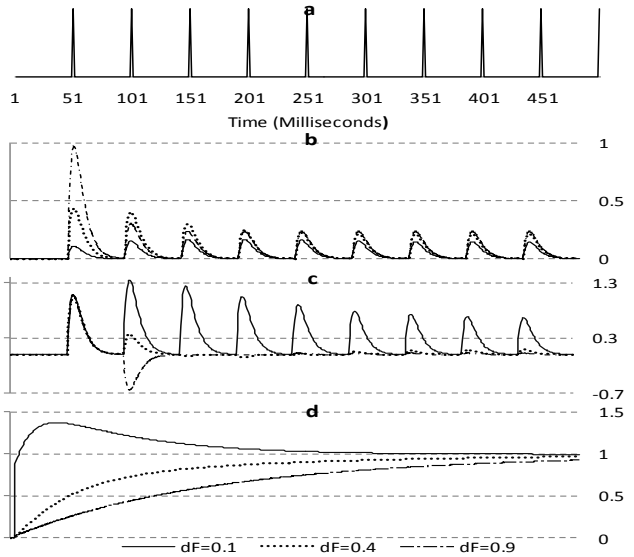

Figure 1 Synapse Temporal Dynamics **a)** Impinging AP. Processing period is 500 milliseconds, and ISI of the AP is 50 milliseconds. **b)** Synaptic PSP for $\Delta F = 0.1$, 0.4 and 0.9. Smaller ΔF leads to potentiating synaptic efficacy, meanwhile higher values of ΔF truns to depressing synapse **c)** Synaptic sensitivity in response to ΔF changes . The synapse sensitivity is a non-linear function of ΔF and AP pattern. For smaller ΔF, the synaptic efficacy represents positive change in response to occurring APs, while for higher ΔF the sensitivity flips through time. **d)** Synaptic efficacy in response to ISI variation for different ΔF. Smaller ΔF has higher sensitivity in lower ISI, and the peak of synaptic efficacy move to lower ***ISI*** for higher values of ΔF.

A. Parameter Sensitivity Analysis

Each element of the state matrix -A_s, A_t- and input vector -B_s, B_t- is a continuous function of the synapse internal parameters. Equation (3) denotes derivative of state vector relative to any of synapse parameters:

$$\frac{\partial X_{n+1}}{\partial P} = \left(\frac{\partial A_s}{\partial P} + 1_{ap}(n) * \frac{\partial A_t}{\partial P}\right) * X_n \ldots$$
$$\ldots + \left(A_s + 1_{ap}(n) * A_t\right) * \frac{\partial X_n}{\partial P} + \left(\frac{\partial B_s}{\partial P} + 1_{ap}(n) * \frac{\partial B_t}{\partial P}\right) \quad (3.a)$$
$$\frac{\partial V_n}{\partial P} = C * \frac{\partial X_n}{\partial P} \quad (3.b)$$

Variable P can be any synapse parameters. Figure (1.b) shows $\frac{\partial V_n}{\partial \Delta F}$ for three different value of facilitation factors. Equation (3) is the key component of the learning, which defines a recursive function for calculating derivative of PSP membrane relative to synapse parameters. Figure (1.c) shows the PSP derivative for different facilitation factors.

B. AP Pattern Sensitivity Analysis

Synaptic efficacy can be quantified by the ratio of vesicle release in the m^{th} and first impinging AP. The ratio for different inter-spike interval (ISI) represents a quantitative measure for synaptic sensitivity to variation in AP pattern. Equation (4) denotes the efficacy ratio for ISI of k-milliseconds, and figure (1.d) shows the ratio for different ISI with different facilitation factors.

$$E_k = X_{m*k+1}(4)/X_{k+1}(4) \quad (4)$$

Similarly, temporal dynamics can be analyzed in response to mean-firing rate. For homogenous firing rate, the synapse

model turns to a linear time-invariant model. Equation (5) represents synapse temporal dynamics and the sensitivity analysis for mean firing of ρ.

$$\bar{X}_{n+1} = (A_s + \rho * A_t) * \bar{X}_n + (B_s + \rho * B_t) \quad (5.a)$$

$$\frac{\partial \bar{X}_{n+1}}{\partial \rho} = (A_s + \rho * A_t) * \frac{\partial \bar{X}_n}{\partial \rho} + A_t * \bar{X}_n \quad (5.b)$$

$$\bar{X}_{\text{steady}} = (I - A_s - \rho * A_t)^{-1} * (B_s + \rho * B_t) \quad (5.c)$$

The next section will discuss about learning algorithm in DSNN.

III. Synaptic adaptation Algorithm

A biological plausible learning rule capable of simultaneously adjusting LTP and STP synaptic parameters is introduced. The learning rule is a supervised learning rule in which the similarity between the network output spike train and a desired spike train is increased. It is shown that the link between the learning mathematics and biological counterpart leads to a reward-based STDP learning rule applicable of training a DSNN independent of its topology and size. In the following sections, similarity measure between spike trains and then supervised learning rule are introduced. The reward-based STDP and its formulation will be described too.

A. Spike Train Similarity Measure

Train similarity measure evaluates the coincidence between APs in the test and desired spike trains [9, 14]. The measure quantifies the similarity between the test and desired spike trains in two different time scales. The fine temporal scale identifies each AP of the test spike train as one of similar, missing or extra categories; meanwhile the global scale measure returns a normalized value determining the overall similarity between two trains. An AP is considered similar if it occurs in time vicinity of one of APs in the desired spike train. This vicinity time interval is called similarity range and determines acceptable jitter between spike pairs in the test and desired spike train- generally, similarity range is less than half of refractory period. The overall similarity of two spike trains are defined by;

$$S = N_s / max(1, N_d, N_t) \quad (6)$$

N_s is the number of similar spikes; N_d and N_t correspondingly denote number of spikes in the desired and test spike trains. Figure 2 represents the similar, missing and extra spikes, in which S gets value of 0.4.

Similarity of two spike trains are increased by reducing number of missing and extra spikes. To increase the similarity, membrane potential at extra spikes needs to be dropped below firing threshold, at the same time membrane potential of missing spikes need to be increased above threshold. The learning algorithm corresponds to membrane potential changes leading to increase the overall similarity between two spike trains.

B. Learning Algorithm

The learning algorithm modifies synaptic parameters to minimize the following objective function:

$$E = \sum_{i=1}^{max(N_d, N_t)} \big(V(n_i) - V_d(n_i)\big)^2 \quad (7)$$

Variable $V(n_i)$ is the neuron membrane potential and variable $V_d(n_i)$ is the desired potential at n_i. In equation (7),

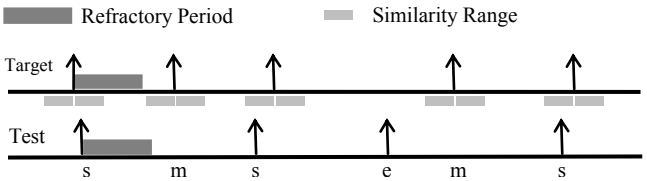

Figure 2 Similar (s), missing (m) and extra (e) spikes. The S value is equal to 0.4.

n_i represents time of similar, missing or extra spikes. $V_d(n_i)$ is equal to $V_{th} + \varepsilon$, $\varepsilon > 0$, at similar and missing spike times, and it is $V_{th} - \varepsilon$ at extra spikes. Though ε is assumed to be a constant value, it is shown that ε can be defined by overall similarity of two trains increasing learning speed [14]. The learning algorithm applies a recursive gradient descent search in synapse parameter space minimizing dissimilarity between the test the target spike trains [14]. The learning algorithm is defined by:

$$\frac{\partial E}{\partial P} = \sum_{i=1}^{max(N_d, N_t)} 2 * \big(V(n_i) - V_d(n_i)\big) * \frac{\partial V(n_i)}{\partial P} \quad (8.a)$$

$$P = P + \eta * \frac{\partial E}{\partial P} \quad (8.b)$$

Both of $V(n_i)$ and $\frac{\partial V(n_i)}{\partial P}$ are recursively updated using equations (2) and (3), and η determines learning rate. The the learning process is recursive updates of equations (2), (3), (8) plus similarity measure process.

C. Reward STDP Learning

The learning algorithm (8.a) is comprised of two terms: a) a positive or negative term, $V(n_i) - V_d(n_i)$- determining temporal performance of the network, b) local gradient of each individual synapse relative to its internal parameters- $\frac{\partial V(n_i)}{\partial P}$.

In reward learning algorithm, each individual unit of a system is rewarded or punished equally depending to the system response to a stimuli. Similarly, the term $V(n_i) - V_d(n_i)$ plays the same rule in supervised learning. In reward learning, the reward $R(n_i)$- equal to $V(n_i) - V_d(n_i)$ in supervised learning- can be generalized to a more sophisticated evaluative function of the network output, and it can be defined without the desired spike train. The gradient term can be rewritten as;

$$T_n = \frac{\partial X_{n+1}}{\partial P} = \left(\frac{\partial A_s}{\partial P} * X_n + A_s * \frac{\partial X_n}{\partial P} + \frac{\partial B_s}{\partial P}\right) + 1_{ap}(n) \ldots$$
$$\ldots * \left(\frac{\partial A_t}{\partial P} * X_n + A_t * \frac{\partial X_n}{\partial P} + \frac{\partial B_t}{\partial P}\right) = T_{s,n} + 1_{ap}(n) * T_{t,n} \quad (9)$$

T_n is a defined by the history of impinging APs and it is not limited to only a pair of pre and post synaptic APs.

The learning rule is defined by two factors: a) T_n a function of pre-synaptic AP time and b) $R(n_i)$ reward function. The reward-based STDP learning rule is defined by;

$$P = P + \eta * \sum_{i=1}^{N} R(i) * T_i \quad (10)$$

In contrast to standard STDP learning rule [2], the reward-based STDP is approximately a causal learning rule. Time period of non-causal term is only limited to delay in returning reward; where for the proposed similarity measure the delay is equal to similarity range. Thus, the reward-based STDP can be an online learning rule. Reward-based STDP learning rule adjusts synaptic parameters based on synapse local activity regulated by overall performance of the

network. Thus, the learning rule is applicable in larger DSNN independent of the network topology and layers.

IV. Application

This section will address a specific application of synapse model in building neural model for generating different spiking and bursting pattern of the rat cortical neurons [15]. The neuron only consists two synaptic connections; the first synapse transfers the incoming spike train to the neuron and the second one projects the neuron output to itself. ISI of the input spike train is a 50 milliseconds, and the objective is generating different spiking patterns reported in [15]. There are only four free parameters in the model; ΔF and N_{max} for each synaptic connection are adjusted by the learning algorithm. Figure 3 shows regular spiking (RS) and chattering spike (CH) neuron output [15] generated by neural model in response to the impinging APs.

In contrast to mathematical model for spike generation [15], temporal dynamics of synapses and neuron in this neural model has a clear biological interpretation, meanwhile the processing is linear.

V. Conclusion

Two fundamental steps proposed in building biological spatio-temporal computational unit. A linear synapse model for STP and LTP was introduced, and a generalized reward-based STDP learning rule applicable in DSNN was defined. The proposed model builds a novel biological computation unit capable for neural simulation and temporal signal processing.

The main question is the convergence speed and extension of the learning rule for large scale DSNN. Both issues plus its application in brain cortical simulation and neural-engineering will be addressed in the coming paper.

Appendix A

This appendix defines elements of state matrix and input vector for the linearized synapse model. Through linearization process the higher term of F^2N is set to zero; other term such as $1/(\tau_F * \tau_R)$ are set to zero too. Both of approximations are valid assumption, because of large time constants and strong contribution of $F * N$ term in synapse dynamics.

Table 1 Nominal values for model parameters

Parameter		Value
$\boldsymbol{\tau_F}$	Facilitation Time Constant	100 msec
$\boldsymbol{F_0}$	Resting Facilitation	0…1
$\boldsymbol{\Delta F_0}$	Facilitation Increment Factor	0…1
$\boldsymbol{\tau_R}$	Vesicle Recovery Time Constant	200 msec
$\boldsymbol{N_{max}}$	Maximum Number of Release Sites	1...1000
$\boldsymbol{\tau_G}$	Rise time of PSP	5 msec
$\boldsymbol{\tau_K}$	Rise time of PSP	5 msec

$F_{n+1} = A_{11,n} * F_n + B_{1,n}$

- $A_{11,n} = \left(1 - \frac{1}{\tau_F}\right) - \Delta F * 1_{ap}(n)$ $\quad B_{1,n} = \frac{1}{\tau_F} * F_0 + \Delta F * 1_{ap}(n)$

$N_{n+1} = A_{22,n} * N_n + A_{23,n} * F_n * N_n + B_{2,n}$

- $A_{22,n} = \left(1 - \frac{1}{\tau_R}\right) - \left(\Delta F + \frac{1}{\tau_F} * F_0\right) * 1_{ap}(n)$
- $A_{23,n} = \left(\frac{1}{\tau_F} + \Delta F - 1\right) * 1_{ap}(n)$ $\quad B_{2,n} = \frac{1}{\tau_R}$

$R_{n+1} = F_{n+1}N_{n+1} = A_{31,n} * F_n + A_{32,n} * N_n + A_{33,n} * F_n * N_n + B_{3,n}$

- $A_{31,n} \cong \frac{1}{\tau_R} - \frac{\Delta F}{\tau_R} * 1_{ap}(n)$
- $A_{32,n} \cong \frac{F_0}{\tau_F} + \left[\Delta F - \Delta F^2 - \frac{\Delta F}{\tau_R} - 2 * \frac{\Delta F * F_0}{\tau_F}\right] * 1_{ap}(n)$
- $A_{33,n} = \left(1 - \frac{1}{\tau_F} - \frac{1}{\tau_R}\right) + \left[2 * \Delta F^2 - 3 * \Delta F + 2 * \frac{\Delta F - F_0 + \Delta F * F_0}{\tau_F} + \frac{\Delta F}{\tau_R}\right] * 1_{ap}(n)$
- $B_{3,n} = \frac{\Delta F}{\tau_R} * 1_{ap}(n)$

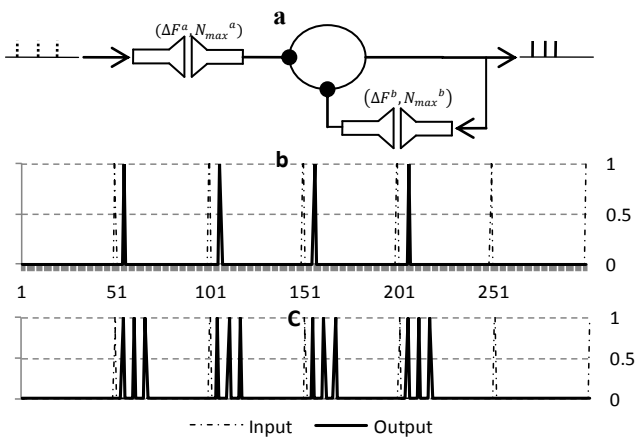

Figure 3 Neural model for rat cortical neuron **a)** neural model structure **b)** RS pattern, $(\Delta F^a, N_{max}{}^a) = (0.36,39)$, $(\Delta F^b, N_{max}{}^b) = (0.21,44)$ **c)** CH pattern, $(\Delta F^a, N_{max}{}^a) = (0.33,56)$ and $(\Delta F^b, N_{max}{}^b) = (0.05,54)$

$G_{n+1} = A_{42,n} * N_n + A_{43,n} * F_n * N_n + A_{44,n} * G_n$

- $A_{42,n} = N_{max} * \left(\frac{1}{\tau_F} * F_0 + \Delta F\right) * 1_{ap}(n)$
- $A_{43,n} = N_{max} * \left(1 - \frac{1}{\tau_F} - \Delta F\right) * 1_{ap}(n)$ $\quad A_{44,n} = \left(1 - \frac{1}{\tau_G}\right)$

$K_{n+1} = A_{52,n} * N_n + A_{53,n} * F_n * N_n + A_{55,n} * K_n$

- $A_{52,n} = N_{max} * \left(\frac{1}{\tau_F} * F_0 + \Delta F\right) * 1_{ap}(n)$

$A_{53,n} = N_{max} * \left(1 - \frac{1}{\tau_F} - \Delta F\right) * 1_{ap}(n)$ $\quad A_{55,n} = \left(1 - \frac{1}{\tau_K}\right)$

References

[1] L. F. Abbott, and Wade G. Regehr, "Synaptic computation," NATURE, Vol 431, Oct 2004

[2] Abigail Morrison, Markus Diesmann, Wulfram Gerstner, "Phenomenological models of synaptic plasticity based on spike timing, " Biol Cybern, 2008, 98:459–478

[3] Thomas Natschlager,Wolfgang Maass, Anthony Zador, "Efficient Temporal Processing with Biologically Realistic Dynamic Synapses, " Network, Feb. 2001;12(1):75-87

[4] Jim-Shih Liaw, Theodore W. Berger, "Dynamic synapse: Harnessing the computing power of synaptic dynamics, " Neurocomputing 26-27, 1999, Pages 199-206

[5] Misha Tsodyks, Klaus Pawelzik, Henry Markram, "Neural Networks with Dynamic Synapses, " Neural Computation 10, 821–835, 1998

[6] Jeremy S. Dittman, Anatol C. Kreitzer, Wade G. Regehr, "Interplay between Facilitation, Depression, and Residual Calcium at Three Presynaptic Terminals, " The Journal of Neuroscience, February 15, 2000, *20*(4):1374–1385

[7] Gianluigi Mongillo, Omri Barak, Misha Tsodyks, "Synaptic Theory of Working Memory, " Science 319, 1543, 2008

[8] Jim-Shih Liaw and Theodore W. Berger, " Dynamic Synapse: A New Concept of Neural Representation and Computation, " HIPPOCAMPUS 6:591–600 (1996)

[9] Thomas Kreuz, Julie S. Haas, Alice Morelli, Henry D.I. Abarbanel, Antonio Politi, "Measuring spike train synchrony, " Journal of Neuroscience Methods 165 (2007) 151–161

[10] Robert Gutig, Haim Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions, ", Nature Neuroscience, 2006

[11] Filip Ponulak, Andrzej Kasinski, "Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting, " Neural Computation 22, 467–510 (2010)

[12] Sander Marcel Bohte, "SPIKING NEURAL NETWORKS, " PhD Thesis, 2003

[13] Samanwoy Ghosh-Dastidar, Hojjat Adeli, " A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, " Neural Networks 22 (2009) 1419_1431

[14] Yousefi, A.; Dibazar, A.A.;Berger, T.W.; "Supervised learning in a single layer Dynamic Synapses Neural Network, " IJCNN 2011

[15] Eugene M. Izhikevich, "Simple Model of Spiking Neurons, " IEEE transaction in neural network, Vol. 14, No. 6, Nov. 2003

[16] Wulfram Gerstner, Werner Kistler, " Spiking Neuron Models, " CAMBRIDGE, 2006