Implementation of Biologically Plausible Spiking Neural Networks Models on the POEtic Tissue

J. Manuel Moreno[1], Jan Eriksson[2], Javier Iglesias[2,3], and Alessandro E.P. Villa[3]

[1] Technical University of Catalunya, Dept. of Electronic Engineering,
Campus Nord, Building C4, c/Jordi Girona 1-3, 08034-Barcelona, Spain
moreno@eel.upc.edu

[2] Laboratory of Neuroheuristics, Information Systems Department INFORGE,
University of Lausanne, Lausanne, Switzerland
jan@lnh.unil.ch, Javier.Iglesias@unil.ch

[3] INSERM U318, University Joseph-Fourier Grenoble 1, Pavillon B,
CHUG Michallon, BP217, F-38043 Grenoble Cedex 9, France
Alessandro.Villa@ujf-grenoble.fr
http://www.nhrg.org

Abstract. Recent experimental findings appear to confirm that the nature of the states governing synaptic plasticity is discrete rather than continuous. This means that learning models based on discrete dynamics have more chances to provide a ground basis for modelling the underlying mechanisms associated with plasticity processes in the brain. In this paper we shall present the physical implementation of a learning model for Spiking Neural Networks (SNN) that is based on discrete learning variables. After optimizing the model to facilitate its hardware realization it is physically mapped on the POEtic tissue, a flexible hardware platform for the implementation of bio-inspired models. The implementation estimates obtained show that is possible to conceive a large-scale implementation of the model able to handle real-time visual recognition tasks.

1 Introduction

Among the different types of artificial neural networks models that have been investigated during the last decades spiking neural networks have attracted large research efforts [1], [2] because of their biological plausibility and their suitability for a physical hardware implementation. These neural paradigms usually consider a simplified model for the neuron that is based in an integration process for its inputs and the delivery of an output spike when the membrane potential exceeds a given threshold.

Among different learning mechanisms Spike Timing Dependent Plasticity (STDP), i.e., the modification of the synaptic weights depending on the time correlation between pre- and post-synaptic spikes, has raised an increasing interest [3] due to experimental evidence [4] and observations suggesting that synaptic plasticity may be based on discrete dynamics [5].

In this paper we shall consider a spiking neural network model [6] based on STDP learning rules whose learning dynamics is based on discrete variables. This model has demonstrated excellent properties for discriminating dynamic input stimuli in large-scale networks [7].

J. M. Moreno, J. Madrenas, and J. Cosp (Eds.): ICES 2005, LNCS 3637, pp. 188 – 197, 2005.

The rest of the paper is structured as follows: in the next section we shall provide a brief summary of the learning scheme proposed in the considered model. Then we shall provide the hardware implementation of this model and the procedure for its functional validation. The accuracy of the internal variables used in the model is then scaled down to allow for a compact hardware implementation. After validating this optimization the resulting model is implemented using the POEtic tissue, a prototyping platform for bio-inspired models. Finally, the conclusions and our current development work are outlined.

2 A Biologically Inspired SNN Model

The model consists of Leaky Integrate-and-Fire neuromimes connected by synapses with variable weight depending on the time correlation between pre- and post-synaptic spikes. The synaptic potentials are added until their result $V_i(t)$ overcomes a certain threshold, θ. Then a spike is produced, and the membrane value is reset. The simplified equation of the membrane value is:

$$V_i(t+1) = \begin{cases} 0 & when\ S_i(t) = 1 \\ k_{mem} \cdot V_i(t) + \sum J_{ij}(t) & when\ S_i(t) = 0 \end{cases} \tag{1}$$

where $k_{mem}=exp(-\Delta t/\tau_{mem})$, $Vi(t)$ is the value of the membrane and $S_i(t)$ is the state variable which signals the occurrence of a spike. The value of „J_{ij} is the output of each synapse *(ij)* where *j* is the projecting neuron and *i* is the actual neuron.

When a spike occurs in the pre-synaptic neuron, the actual value of the synaptic output J_{ij} is added to the weight of the synapse multiplied by an activation variable *A*. Conversely, if there is no pre-synaptic spike then the output J_{ij} is decremented by a factor k_{syn}. Then, the value of „J_{ij} corresponds to the following equation:

$$J_{ij}(t+1) = \begin{cases} J_{ij}(t) + (w_{RiRj} \cdot A_{RiRj}(t)) & when\ S_j(t) = 1 \\ k_{syn} \cdot J_{ij}(t) & when\ S_j(t) = 0 \end{cases} \tag{2}$$

where R is the type of the neuron, either excitatory or inhibitory.

If the actual neuron is inhibitory, the factor k_{syn} will reset the output of the synapse after a time step; if the actual neuron is excitatory, the update of the synaptic output depends on the projecting neuron and the STDP rule is applied. An inhibitory cell can not influence another inhibitory cell, i.e. assume a synaptic weight of zero between two inhibitory neurons. The basic synaptic strengths are chosen in order to maintain a balanced excitatory/inhibitory activity within the network.

The changes in strength of an excitatory-excitatory synapse depend on the variable *A* which is a function of on an internal variable L_{ij} given by the following equation:

$$L_{ij}(t+1)=k_{act}\cdot L_{ij}(t) + (YD_j(t)\cdot S_i(t)) - (YD_i(t)\cdot S_j(t)) \tag{3}$$

where k_{act} is a kinetic activity factor, which is the same for all the synapses and *YD* is a “learning” decaying variable that depends on the interval between a pre-synaptic spike and a post-synaptic spike. When there is a spike, *YD* reaches its maximum value at the next time step. In the absence of a spike the value of *YD* will be decremented by the kinetic factor $k_{learn,}$ which is the same for all synapses. When a pre-synaptic spike occurs just before a post-synaptic spike, then the variable L_{ij} is increased and the synaptic strength becomes larger, thus corresponding to a potentiation of the synapse. When a pre-synaptic spike occurs just after a post-synaptic spike, the variable L_{ij} is decreased, the synaptic weight is weakened , thus corresponding to a depression of the synapse. For all kind of synapses, except the excitatory-excitatory, the activation variable is always is set to 1.

The network layout was chosen with 80% of excitatory and 20% inhibitory neurons. Each unit was fully connected within a 5x5 neighborhood, i.e. connected to 24 neurons (Fig. 1).

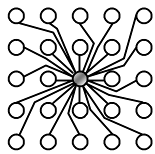


Fig. 1. Connectivity of a single neuron

3 Hardware Implementation

From a structural point of view the SNN model considered in this paper is constituted by four main building blocks: the neuron block, the decay block, the learning block and the synapse block.

The **neuron block** is in charge of implementing the dynamics of the membrane by integrating the pre-synaptic spikes, as indicated in Eq. (1). The characteristics of the parameters of this block are the following:

- The membrane potential has a resolution of 12 bits, with a range [-2048, 2047], and the threshold is kept fixed to +640.
- The membrane decay function has a time constant value of τ=20.
- The refractory period is set to 1 time unit.

The **decay block** will be used in both learning and synapse blocks. This block is aimed to implement a logarithmic decay of the input; it is obtained with a subtraction and controlling the time when it is done depending on the input value. This block is used in many parts of the design and the decaying variable has been labeled *x* in Figure 2. A new value of *x* will be the input of a shift register which is controlled by the most significant bit (*MSB*) of *x* and by an external parameter *mpar*. The output of this shift register will be subtracted from the original value of *x*. This operation will be done when the time control indicates it. The time control is implemented by the value of a counter that is compared with the result of choosing between the external value *step* and the product *(MSB–mpar)·step*. The decay variable τ depends on the input parameters *mpar* and *step*.

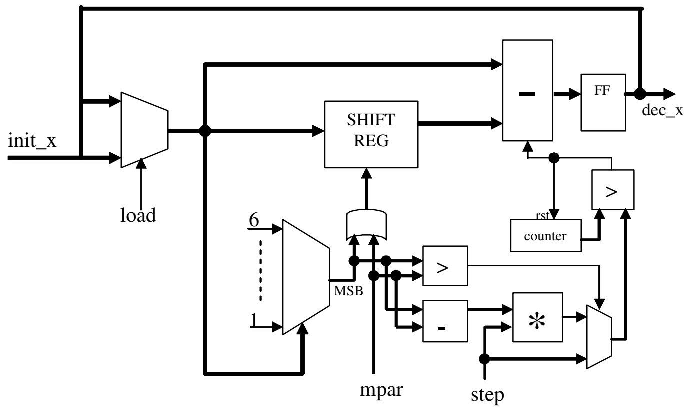


Fig. 2. Diagram of the decay block

The **learning block** “measures” the interval between a spike in the projecting neuron j and the actual neuron i. Depending on these timings and the types of the two neurons, the synaptic strength will be modified.

When a spike is produced by the projecting neuron, the variable YD is set to its maximum value and starts to decay. If a spike is produced by the actual neuron immediately after the presynaptic neuron the value of YD_j is added to the decaying value of L. Conversely, if a spike is produced at first in the actual neuron and later in the projecting neuron, then the value of YD_i is subtracted to the decaying value of L.

If the L variable overcomes a certain threshold L_{th}, positive or negative, then the activation variable A is increased or decreased, respectively, unless the variable had reached its maximum or minimum, respectively. If the variable A is increased, then L is reset to the value $L\text{-}2 \cdot L_{th}$; if A is decreased, then L is reset to $L+2 \cdot L_{th}$.

Figure 3 illustrates the organization of the learning block.

The characteristics of the parameters of the learning block are the following:

- The YD variable has a resolution of 6 bits.
- The time constant for the variable YD is τ=20.
- The learning variable L of 8 bits and $\cdot L_{th}$ is within the range [-128,127].
- The activation variable A is coded by 2 bits and takes four states.
- To improve the sensitivity of the block for long intervals between spikes the time constant for the variable L is set to 4000, but it can be changed depending on the network size implementation.

The **synapse block** is aimed to set the value of J (analogous to the the sum of all post-synaptic membrane potentials) and depends on four factors: the activation level A of the synapse, the spiking state of the projecting neuron S_j and the types of the pre- and post-synaptic neurons (R_i and R_j).

A given weight is set for each synapse. This weight is multiplied by the activation variable A by means of a shift register, such that if A=0, the weight is multiplied by 0, if A=1 it is multiplied by 1, if A=2 it is multiplied by 2, and if A=3 it is multiplied by 4. This weighted output is added to the decaying value of the variable J.

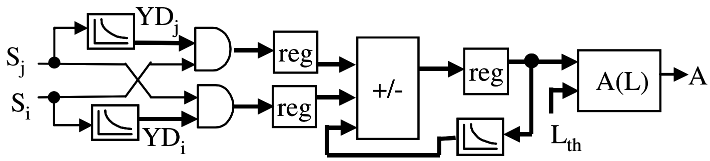


Fig. 3. Diagram of the learning block

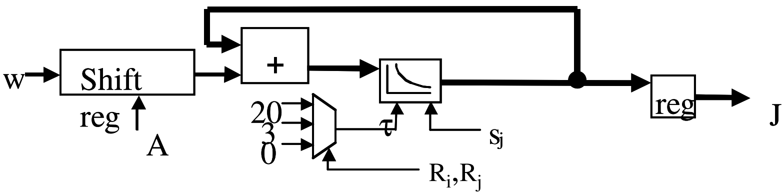


Fig. 4. Diagram of the synapse block

This operation depends on the neuronal types (R_i and R_j). In the current case study there are only two types of neurons, excitatory and inhibitory. If both neurons are inhibitory the weight of the synapse is set to 0 and the value of J is always 0 and no decay is implemented. For the other three types of synapses the time constants are multiplexed, and the multiplexer is controlled by the types of neurons (R_i,R_j). The value of J is obtained at the output of the decay block controlled by the multiplexer. Figure 4 shows the organization of the synapse block.

The characteristics of the parameters of the synapse block are the following:

- The internal resolution of the block is 10 bits, but the output resolution is 8 bits, becasue the internal value of J is divided by 4 to keep the correct scaling with the other parameters.
- The time constants used by this block are listed in Table 1.

Table 1. Time constants for different types of synapses. R=0 corresponds to an excitatory and R=1 to an inhibitory neuron.

Time Constant (τ)	Projecting Neuron Type (R_j)	Actual Neuron Type (R_i)
20	0	0
0	0	1
3	1	0
0	1	1

4 Parameters Tuning

The resolution required to represent the values of the variables and the number of operations to be performed may pose a serious limitation for the final implementation. Therefore, an important step consisted in evaluating the model and tuning its parameters in order to get a satisfactory performance. The implementation used in this study has been based on a neural network of size 15x15 with a connectivity pattern of 24 neurons corresponding to a neighborhood of 5x5 (Fig. 1). The distribution of the 20% inhibitory cells was random. The weights, w, and the initial activation variables, A, were also chosen randomly. Dynamic gradient stimuli have been applied to the neural network. A sequence of vertical bars of gradient intensity move over "strips" of neurons placed in the 2D array of the neural network (Fig. 5).

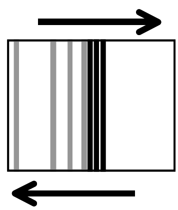
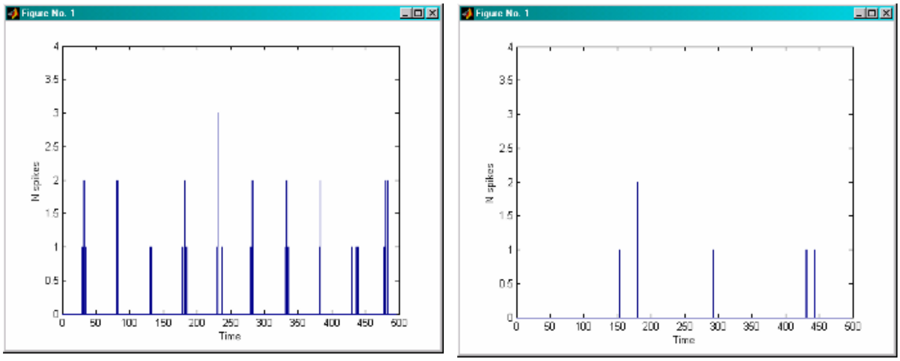


Fig. 5. Input signal applied to the neural network. The arrow to the right means forward sense and the arrow to the left means reverse sense.

The vertical bars may move at different speeds (i.e. spatial frequency). A neuron "hit" by the stimulus receives an input that is proportional to the gradient intensity. The activity of the network has been studied in a "training" condition and in a "test" condition. During training the spatial frequency of the stimulus has been incremented by discrete harmonics (2x, 4x, etc.) in one direction (the "forward" direction). During test, the stimuli were presented in both forward and reverse sense. A Gaussian noise (Mean 0, SD= 48) is applied to all neurons during all the time. The characteristics of the input applied to each neuron are the following:

- T_{CLK}: 20 ns. Maximum amplitude: 127.
- Training period: 20 us. Forward sense
- Test period: 10 us. Forward and Reverse sense

The activity calculated over a "strip" of neurons perpendicular to the direction of the movement represents a measure of "local" activity. In this case, the strip is one-column wide. In Fig. 6 the "local" activity is measured by the count of spikes produced as a function of the time steps. We can observe that in the forward sense there exists an activation pattern with a temporal correlation, but in reverse sense the network output has no such temporal correlation. This result demonstrates that the selected structure of our neural network is able to perform an implicit recognition of dynamic features based on simple unsupervised STDP rules.



(a) (b)

Fig. 6. Local activity in column 1. (a) test stimuli are applied in forward sense. (b) test stimuli are applied in reverse sense.

At a first attempt the resolution of the parameters has been reduced by 2 bits and some values and time constants have been changed to keep the correct scaling. Table 2 shows the new values of the internal parameters after this optimization process. The final organization resulting from this optimization process is depicted in Fig. 7. The simplified model resulting from this optimization has been validated again using the same input stimuli presented in Fig. 5. The results of these simulations demonstrate that the model is still capable of discriminating the input stimuli applied in the forward and in the reverse directions.

Due to the complexity of the design, the simplification of the model is very important to avoid redundancy or to use just the necessary components. For this reason, a further simplification of all the building blocks that constitute the model has been performed [8].

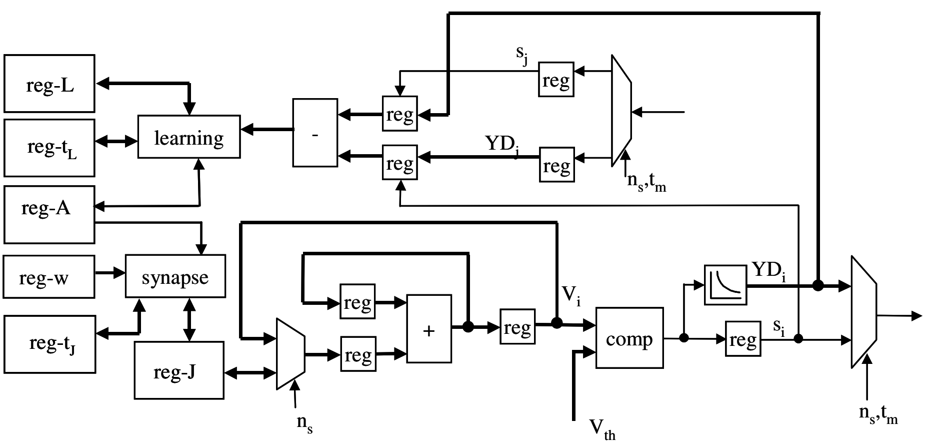


Fig. 7. Block diagram for the serial implementation of the neuron model

Table 2. Resolution of the parameters for an optimized implementation

Parameter	New value
Membrane resolution	10 bits
Threshold	+160
Input (J) resolution	6 bits
Weights (R_i,R_j) (00, 01, 10, 11)	[0:8], [64:128], [128:256], [0:0]
YD resolution	4 bits
L resolution	6 bits
Membrane decay time constant	20
YD decay time constant	20
L decay time constant	4000
J_{R_i,R_j} decay time constants (R_i,R_j) (00, 01, 10, 11)	(20, 0, 3, 0) *values not optimized*

5 Implementation on the POEtic Tissue

The POEtic tissue [9] constitutes a flexible hardware substrate that has been specifically conceived in order to permit the efficient implementation of bio-inspired models. The tissue may be constructed as a regular array composed of POEtic chips, each of them integrating a custom 32-bit RISC microprocessor and a custom FPGA with dynamic routing capabilities.

The custom FPGA included in the POEtic chip is composed of a bi-dimensional array of elementary programmable elements, called molecules. Each molecule contains a flip-flop, a 16-bit lookup table (LUT) and a switchbox that permits to establish programmable connections between molecules.

After the optimization carried out on the neural model in order to facilitate its hardware realization it has been mapped on to the molecules that constitute the POEtic device. The molecule organization shown in Fig. 8 corresponds to the actual structure of the FPGA present in the POEtic device, which is arranged as an 8x18 array of molecules.

The VHDL models developed for the POEtic tissue have been configured and simulated to validate the functionality of the neuron model designed above. After this validation stage the strategy for the simulation of large-scale SNN models has been considered. Since in its actual implementation the POEtic chip only allows for the implementation of a single neuron and the current number of POEtic chips is far less than 10,000 it will be necessary to use a smaller array of POEtic chips whose functionality should be time–multiplexed in order to emulate the entire network. This means that every POEtic chip should be able to manage a local memory in charge of storing the weights and learning variables corresponding to the different neurons it is emulating in time.

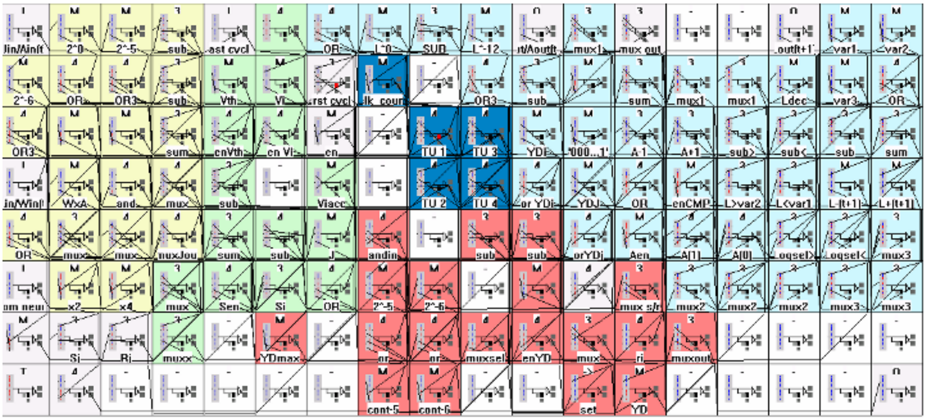


Fig. 8. Molecule-level implementation of the SNN model

A 16-neurons network organized as a 4x4 array has been constructed using this principle. This would permit the emulation of a 10,000-neurons network in 625 multiplexing cycles. Bearing in mind that each neuron is able to complete a time step in 150 clock cycles, this means that the minimum clock frequency required to handle input stimuli in real time (i.e., to process visual input stimuli at 50 frames/second) is around 5 MHz far within the possibilities of the actual clock frequency achieved by the POEtic tissue (between 50 MHz and 100 MHz).

The visual stimuli will come from an OmniVision OV5017 monochrome 384x288 CMOS digital camera. Specific VHDL and C code have been developed in order to manage the digital images coming from the camera. To test the application, artificial image sequences have been generated on a display and then captured by the camera for its processing by the network.

6 Conclusions

In this paper we have considered an unsupervised model for modifiable synapses in a Spiking Neural Network based on discrete interval variables. This model has demonstrated a good performance when used for learning and recognition tasks that involve dynamic input stimuli.

The basic parameters that define the model dynamics have been optimized in order to provide a hardware friendly implementation. The resulting model has been implemented in the POEtic tissue, a flexible hardware platform conceived for the physical realization of bio-inspired models. The results of the current implementation demonstrate that the proposed approach is capable of supporting real-time needs of large-scale spiking neural networks models.

Our current work is concentrated on the physical implementation of the real-time image recognition tasks using the development boards that have been constructed for the POEtic tissue.

Acknowledgements

The work presented in this paper has been funded by the grant IST-2000-28027 (POEtic) of the European Community and by grant OFES 00.0529-2 of the Swiss government. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References

1. Maas, W.: Networks of Spiking Neurons: The Third Generation of Neural Network Models. *Neural Networks* 10 (1997) 1659–1671.
2. Hill, S.L., Villa, A.E.P.: Dynamic transitions in global network activity influenced by the balance of excitation and inhibition. *Network: Computation in Neural Systems* 8 (1997) 165-184.
3. Abbott, L.F., Nelson, S.B.: Synaptic plasticity: taming the beast. *Nature Neuroscience* 3 (2000) 1178–1183.
4. Bell, C.C., Han, V.Z., Sugawara, Y., Grant, K.: Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature* 387 (1997) 278–281.
5. Montgomery, J.M., Madison, D.V.: Discrete synaptic states define a major mechanism of synapse plasticity. *Trends in Neurosciences* 27 (2004) 744-750.
6. Eriksson, J., Torres, O., Mitchell, A., Tucker, G., Lindsay, K., Halliday, D., Rosenberg, J., Moreno, J.M., Villa, A.E.P.: Spiking Neural Networks for Reconfigurable POEtic Tissue. Evolvable Systems: From Biology to hardware. *Lecture Notes in Computer Science* 2606 (2003) 165-173.
7. Iglesias, J., Eriksson, J., Grize, F., Tomassini, M., Villa, A.E.P.: Dynamics of pruning in simulated large-scale spiking neural networks. *Biosystems* Vol. 79 (2005) 11-20.
8. Torres, O., Eriksson, J., Moreno, J.M., Villa, A.E.P.: Hardware optimization and serial implementation of a novel spiking neuron model for the POEtic tissue. *BioSystems* 76 (2003) 201–208.
9. Moreno, J.M., Thoma, Y., Sanchez, E., Torres, O., Tempesti, G.: Hardware Realization of a Bio-inspired POEtic Tissue. *Proceedings of the NASA/DoD Conference on Evolvable Hardware*. IEEE Computer Society (2004) 237-244.