Synaptic Control for Hardware Implementation of Spike Timing Dependent Plasticity

Salah Daddinounou, Elena Ioana Vatajelu

Synaptic Control for Hardware Implementation of Spike Timing Dependent Plasticity

Salah DADDINOUNOU, Elena Ioana VATAJELU
Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France*
salah.daddinounou@univ-grenoble-alpes.fr, elena-ioana.vatajelu@cnrs.fr

***Abstract*—Spiking neural networks (SNN) are biologically plausible networks. Compared to formal neural networks, they come with huge benefits related to their asynchronous processing and massively parallel architecture. Recent developments in neuromorphics aim to implement these SNNs in hardware to fully exploit their potential in terms of low energy consumption. In this paper, the plasticity of a multi-state conductance synapse in SNN is shown. The synapse is a compound of multiple Magnetic Tunnel Junction (MTJ) devices connected in parallel. The network performs learning by potentiation and depression of the synapses. In this paper we show how these two mechanisms can be obtained in hardware-implemented SNNs. We present a methodology to achieve the Spike Timing Dependent Plasticity (STDP) learning rule in hardware by carefully engineering the post- and pre-synaptic signals. We demonstrate synaptic plasticity as a function of the relative spiking time of input and output neurons only.**

***Index Terms*—Spiking Neural Networks (SNN), unsupervised learning, Magnetic Tunnel Junction (MTJ), neuromorphics.**

I. Introduction

The rapid growth of Artificial Intelligence (AI) has enabled ubiquitous use of smart applications like image recognition [1], sensing [2] and decision making. Using Artificial Neural Networks (ANN) for AI comes with a certain cost, mainly energy & time. Indeed, software-based ANNs which are running on classic Von Neuman computers (separated CPU and RAM) suffer from expensive data shuffling between the memory and the processor. In addition, formal ANNs are not optimal because the signal circulates continuously between the processing unit and the memory, leading to high energy consumption [3]. This challenge has pushed researchers to take inspiration from the biological nervous systems to solve the problem. The biological nervous system is by far the most energy efficient computer [3]. For example, training a state-of-the art natural language processing model on a modern supercomputer consumes 1000 kWh, which is the energy consumed by a human brain for the entirety of its tasks over a duration of six years [3].

Neuroscience research states that signals in the brain circulate in the form of spikes between neurons via synapses [4]. A first step towards highly energy efficient ANNs is to move from formal NNs to Spiking Neural Networks (SNNs) in which all signals are coded and transmitted as spikes. Due to this special signal coding, the traditional training algorithms such as linear regression or backpropagation become incompatible, so new training algorithms have to be considered. Looking again to neuroscience, we find the theory which states that "cells that fire together wire together" [5]. This causality theory is the bases of the Spike Timing Dependent Plasticity (STDP) learning rule which is suitable for training SNNs. The STDP learning process consists in adjusting the connection strengths (i.e., synaptic weights) based on the relative timing of a particular neuron's output and input states. A hardware implementation of an SNN trained by STDP requires neurons able to emit spiking signals (the Leaky-Integrate and Fire - LIF neuron for example) and synapses able to adjust their weights in real-time during learning (i.e., have plasticity) and retain the value of the trained weight (i.e., have memory). Although the applications of AI based on neural networks are widely spread, the solutions for physically implemented spiking neural networks (SNNs) with on-line learning are still rare. In addition, the proposed solutions are not thoroughly analysed in terms of quality and robustness.

To the best of our knowledge, this paper is the first work that studies the control of probabilistic MTJ-based synapses to obtain the STDP. Using the proposed signal profile for input and output spikes we achieve learning in time domain, as it is found in biology (i.e., the magnetic synapse changes its conductance level only according to the time delay between the spikes of its corresponding neurons). In this paper we focus only on the learning process and synaptic plasticity, the design and evaluation of the spiking neurons are out of our scope.

The rest of the paper is organised as follows: in section II preliminary notions are introduced, to describe the operation of the magnetic tunnel junction (MTJ) and the operation of a spiking neural network (SNN); Section III briefly summarises the related state of the art; Section IV presents the proposed methodology, while Section V describes the obtained results; Section VI concludes the paper.

II. Preliminary

A. Magnetic Tunnel Junction MTJ

A Magnetic Tunnel Junction (MTJ) consists of a stack of two magnetic layers with different thicknesses separated by a thin oxide layer. The magnetization of the thicker magnetic layer is fixed, whereas the magnetisation of the thinner layer is

[*]Institut National Polytechnique Grenoble Alpes

This work is founded by the Agnece National de la Recherche - France within the EMINENT Project ANR-19-CE24-0001

free to be aligned or opposite to the fixed one. The orientation of the free layer changes depending on the direction of the writing current which should be higher in absolute value than a certain threshold I_{c0}. The MTJ has a low resistance when the two magnetic layers are parallel, it has a high resistance when the two magnetic layers are anti-parallel. This device having two stable resistances is used as a non-volatile (NV) memory with high resistance (logic 1) and low resistance (logic 0). Figure 1 shows a schematic of an MTJ and the switching between the two resistance states.

We use a VerilogA MTJ model [6] that allows a wide range of user-defined geometric and magnetic parameters including the MTJ dimensions, saturation magnetization, damping factor, resistance area product, crystalline anisotropy and temperature. The MTJ model captures the differential Landau-Lifshitz-Gilbert (LLG) equation which describes the precessional motion of a magnetization vector.

Spin Transfer Torque (STT) is the mechanism by which the MTJ switches between the two resistance states. The switching mechanism is stochastic by nature due to thermal fluctuation of magnetization. It is responsible for the large fluctuations in the switching duration. The dynamic model mainly calculates the average switching delay τ_{sw} (with 50% of switching probability). Depending on the magnitude of switching current, the dynamic behavior of MTJ can be divided into two regimes: Sun model ($I > I_{c0}$) and Neel-brown model ($I < 0.8I_{c0}$) where I is the current across the MTJ and I_{c0} is the coercive current. The Sun model is also called precessional switching regime which addresses fast switching (switching time lower than 3ns) but it has high energy consumption due to the high current density. Reversely, in the Neel-Brown regime the switching is slow but with low power consumption due to low current density; this regime is referred to as thermally assisted switching.

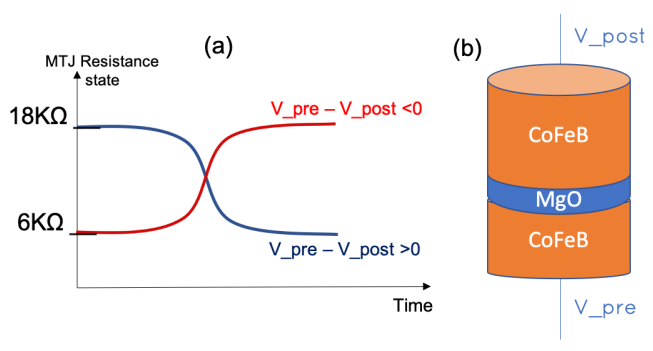

Fig. 1. (a) The MTJ resistance as a function of time during switching (the MTJ has only two stable states). (b) A schematic of MTJ, the upper layer's magnetisation is pinned, whereas the lower one is controlled by the value of the current passing through it and its direction. V_{pre} and V_{post} represent the control signals, which in the case of the SNN under study are the spikes of the input and the output neurons respectively.

B. Spiking Neural Networks

SNN is a network of input spiking neurons (presynaptic) and output spiking neurons (postsynaptic) fully connected by synapses through which the neuronal information is conveyed. The synapses can be organized in a crossbar array, where the input and output neurons are at one extremity of each row and column respectively Fig. 2b. The communication between neurons is achieved by trains of spikes. The input information is spike-coded and fed to the input neurons. For example, in an image recognition task, each pixel has a corresponding input neuron, and the spiking frequency (or rate) of that input neuron is proportional to the grayscale intensity of the pixel that it represents. The network is trained to learn the task (recognition) by adapting the synaptic strengths following the local learning rule of STDP.

In this paper we analyse the behavior of a hardware-implemented SNN with probabilistic MTJ synapses, the synaptic weight is coded in conductance levels, and a local training algorithm based on STDP is used.

C. The MTJ-based Synapse

In the SNN under study, MTJ-based synapses are used. The synaptic weights are expressed in conductance levels. Various conductance levels for a single synapse are obtained by connecting multiple MTJs. Indeed, a single MTJ device is capable of only 2 conductance levels, but by connecting several MTJ devices in series or parallel, multiple conductance levels can be achieved. Here, we propose a compound synapse designed with multiple MTJs connected in parallel. For demonstration purpose only, in this paper we study a synapse built with 4 MTJs connected in parallel as shown in Fig. 2a, the two terminals V_{pre} and V_{post} are the spiking voltages of presynaptic and postsynaptic neurons respectively. Four MTJs per synapse is largely sufficient to learn simple datasets like MNIST, this latter has even already been trained with only two states synapses using binarized neural networks [7]. By using (n) number of MTJs in parallel, we get (n+1) levels of conductance in the synapse. This synapse is capable of 5 conductance levels, the smaller conductance is achieved when all MTJs are in anti-parallel orientation (high restive state) and the larger conductance is achieved when all MTJs are in parallel orientation (low restive state). Intermediary conductance values are obtained when some MTJs are in anti-parallel state while the others are in parallel state.

In order to achieve the different conductance levels, the MTJs should be able to switch their states independently. This condition can be met by controlling the MTJs in the probabilistic region, by taking advantage of the thermal switching. This approach is the bases of our proposed methodology and will be described in detail in section IV.

D. The Training Algorithm

We use an adapted STDP rule to be implementable in hardware, it works as follows: when an input neuron spikes, the signal is transmitted to all the outputs in the bottom of each column via the synaptic connections. As a result, output neurons accumulate the incoming signals. Once one of them reaches the threshold, it fires. Many of them may reach the threshold and fire. During training, the connections between

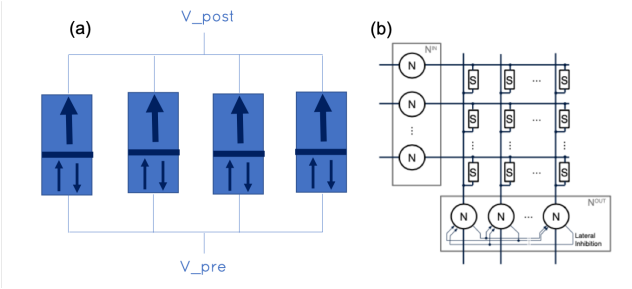

Fig. 2. (a) Schematic of a compound synapse with 4 MTJs in parallel, (b) schematic of the single layer SNN implement as a synapse crossbar array showing also the input and the output neurons

the responsible input neuron and the firing outputs should be adjusted consequently, in two ways:

- *Potentiation:* If the output neuron fires immediately after the input neuron, the corresponding synapse's weight (its conductance in our case) is increased. The early the output fires, the bigger the increase of its synaptic weight. The delay between output spike and input spike gives the importance of causality, i.e., smaller the delay, larger the causality.
- *Depression:* when the relative input-output spiking time is very long, it means that the input neuron was not the direct responsible of activating that specific output, this lack of causality leads to penalising the output neuron by decreasing its connected synaptic connection. The larger the delay of output firing, the larger the decrease of its synaptic weight.

III. Related work

The state of the art of the SNN hardware-implementations is very large as different approaches exist and multiple implementations are being investigated. The authors of [8] argue that there is a big discrepancy between the promise of efficient computing with SNNs and the actual implementations on currently available computing hardware. This is due to the fact that the highly parallel structure, sparse communication, and in-memory computation proposed by SNNs stands in contrast to the sequential and central processing of data constrained by the memory wall between processor and memory on CPUs and GPUs. To mitigate this problem, massively parallel digital architectures of SNN were proposed, among the most eminent ones are TrueNorth [9] from IBM and SpiNNaker [10] from the University of Manchester. TrueNorth is made of 1 million neurons connected by 256 million synapses. The chip is trained off-line so is used only for inference. The test accuracy reaches 99.4% on MNIST handwritten digits database, with an energy consumption of $108\mu J$ per image. SpiNNaker chip on the other hand contains 18 cores with approximately 1K neurons and 1K synapse by core, it demonstrated 95% accuracy consuming $3.3mJ$ per image.

A most disruptive implementation of SNN came with the rise of the emerging non-volatile memories. First, memristors based on conducting filaments, phase change materials and magnetic tunnel junction were demonstrated fitted to stand as synapses thanks to their non-volatility (to retain the synaptic wight) and their adjustable conductance (for plasticity). Later, the memristors were demonstrated fitted for the neurons as well. The authors of [11] demonstrated a system level network with synapses where one synapse is made of one memrestive device based on the conductive filament technology, whereas they use a CMOS based neurons which feature inhibition and homeostasis. In nominal case, the network reaches comparable efficiency with the state of the art when evaluated on MNIST database (93.5 % with 300 output neurons). And has an immunity against variability, where a relative standard deviation of 50 % is tolerated in all device parameters.

In [12] the authors propose an All-Spin Artificial Neural Network (ASANN) which uses spintronic devices to implement both synapses and neurons. The authors propose a Compound Spintronic Synapse (CSS) made of multiple vertically stacked MTJs (2N resistance states can be achieved by stacking N MTJs). They also proposed a compound spintronic neuron (CSN) enabling a multi-step transfer function. The network performs off-line learning paradigm, then the calculated synaptic weights are mapped into CSSs discrete resistance states. The standard back-propagation algorithm was employed for the training process. Many works are focused on the training of SNNs. For instance, [13] presents a network for digit recognition which is based on mechanisms with increased biological plausibility, i.e., conductance-based instead of current-based synapses, STDP with time-dependent weight change, lateral inhibition, and an adaptive spiking threshold. Using this unsupervised learning scheme, their architecture achieves 95% accuracy on the MNIST benchmark.

Most of the proposed techniques to implement SNN in hardware are capable of inference only. Whereas it is the on-line training which should be tackled in order to reduce energy consumption. The authors of [14] propose a toy model of a post-synaptic neuron spiking as a response to the activity of two pre-synaptic neurons, then the memrestive synapses update their state variable consequently, but in one direction only. Their training rule is lacking time dependence also, the output neuron spikes whenever it receives an activity. A similar implementation was proposed in [15] using different CMOS neurons for input and output, with a memristive synapse. A postsynaptic neuron potentiates simultaneously the responsible synapse and depresses all the others systematically. STDP time dependence is again not properly implemented. The novelty of our work is the proper implementation of STDP with a clear time dependence which can be used in hardware-implemented SNN. This will allow online local learning with a compound spintronic synapses (synaptic weight is encoded as conductance). We show how, by carefully choosing the signal shape to emulate the pre- and post-synaptic spikes, the STDP can be implemented in hardware.

IV. Methodology

We focus in this study only on the synapse and we analyse its behavior during the network training process. Our goal is

to design a signal profile for the pre- and post-synaptic spikes respectively, which allow for the implementation of the STDP learning. We carried out electrical simulations of the synapse presented in Fig. 2a, where the signals V_{pre} and V_{post} are chosen so that the voltage drop $V_{pre} - V_{post}$ demonstrates the desired behaviour, i.e., it can change the conductance of the synapse to emulate time-dependent potentiation and depression. In order to tune the synapse conductance, the MTJs should be able to switch their states independently. This condition can be met by controlling the MTJs in the probabilistic region. The probability of MTJ switching depends directly on the applied voltage: amplitude and pulse width. In other words, we can use the MTJ stochasticity to our advantage and control the compound synapse in such a way, that when applying a voltage pulse of a certain amplitude, each MTJ will have a different switching delay. In this case, if the synapse is at minimum conduction level (all MTJs in high resistive state) there exist a positive voltage of amplitude V_{prob} which, when applied across the synapse, will cause the MTJs to switch one by one to the low-resitive state. In this way, the conductance of the synapse will gradually increase to its highest value (all MTJs in low resistive state), thus effectively emulating the synaptic potentiation. In a similar manner, the depression can also be emulated. The purpose of this work is to identify the value of V_{prob} for synaptic potentiation and depression and to optimise it for STDP learning. Network training consists of adjusting the synaptic weights (i.e., the conductance). In section II.D we described the basics of the STDP rule where the important parameter is the relative spiking time between input and output neurons. In order to reproduce this behaviour for the spintronic compound synapse under study, the voltage drop $V_{pre} - V_{post}$ across the MTJs should be delay-dependent as shown in Fig. 3. As the pre- to post-synaptic spiking delay increases, we distinguish five behaviours in this order:

1) *High potentiation:* If the postsynaptic neuron spikes immediately after the presynaptic neuron, the voltage drop $V_{pre} - V_{post} > 0$ is maximum so the conductance of the synapse is raised significantly.
2) *Low potentiation:* The voltage drop still positive but with smaller amplitude as the delay increases, the conductance of the synapse is then raised by a small amount accordingly.
3) *Unchanged conductance:* This is a transition region between potentiation and depression, the voltage drop decreases, it can no longer perform potentiation. It decreases more as the time delay increases, it becomes negative, but dose not perform depression either because it doesn't exceed the negative threshold voltage yet.
4) *Low depression:* The large delay allows the voltage drop $V_{pre} - V_{post}$ to have a negative amplitude which is enough to lower the conductance level of the synapse.
5) *High depression:* If the postsynaptic pulse arrives very late compared to the presynaptic pulse, the voltage drop will have a very large negative amplitude, the synaptic connection is then penalized by lowering its conductance drastically.

The Fig. 3 shows how the plasticity in the synapse can be obtained by tailoring the shapes of presynaptic and postsynaptic pulses. The V_{pre} pulse has a triangular shape with a positive and a negative part, this decreasing voltage shape converts the increasing delay into a gradual decrease in voltage drop.

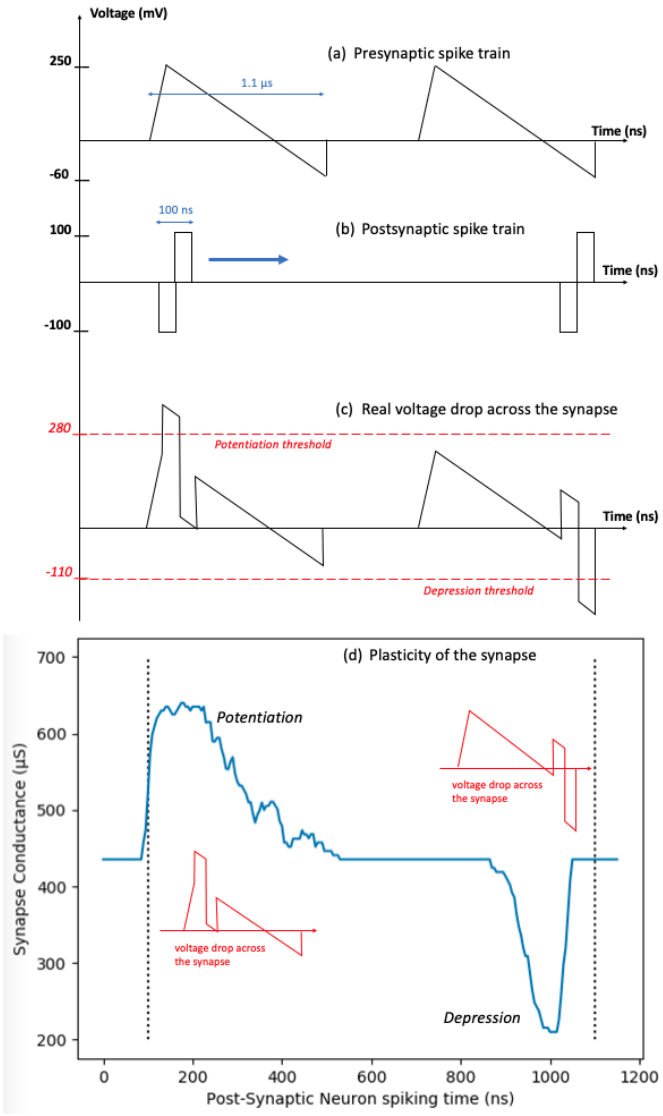

Fig. 3. (a)The presynaptic neuron signal, The pulse V_{pre} has a triangular shape of $1\mu s$. (b) The postsynaptic neuron signal, The pulse V_{post} has positive and negative rectangular parts with a total duration of $100ns$. (c)*left:* The voltage drop $\mathbf{V_{pre}}$-$\mathbf{V_{post}}$ across the synapse is positive and rises above the threshold when V_{post} arrives short after V_{pre} resulting in a potentiation. (c)*right:* The voltage drop $\mathbf{V_{pre}}$-$\mathbf{V_{post}}$ across the synapse is negative and falls under the threshold when V_{post} arrives at the end of V_{pre} pulse, resulting in a depression. (d) The average conductance of the synapse over 20 simulations for each position of the V_{post}. The V_{pre} is fix, it always starts at $100ns$, it is delimited by the two vertical lines. Whereas V_{post} comes at diffident times (before, during and after V_{pre}) the conductance is probed only at the end of each simulation.

The signal transmitting the input spike to the synaptic array has a duration of $1\mu s$, a maximum positive amplitude of

$210mV$ and a maximum negative amplitude of $60mV$. These values were chosen after a thorough study, the details will be laid out in the next section. The V_{post} pulse on the other hand has two rectangular parts with amplitudes $-100mV$ and $100mV$ respectively, each with $50ns$ pulse width.

First, when only V_{pre} or V_{post} spikes, the voltage drop across the synapse is not large enough to modify the synaptic weight (necessary condition for network during inference). Training requires both input and output neurons to spike in order to update the weight of the connected synapse. When a synapse sees both pre- and post- synaptic signal, the voltage drop across the synapse is given by $V_{pre}-V_{post}$. Therefore, the negative amplitude of V_{post} adds up to the positive amplitude of V_{pre}. If the positive threshold is exceeded then the synapse will potentiate. If the output neuron spikes late, and the post-synaptic pulse arrives towards the end of the pre-synaptic pulse, the positive part of V_{post} is subtracted from the negative part of V_{pre}. If the negative threshold is exceeded the synapse will depress. This way, the potentiation and depression in the synapse, is only decided by the time delay between input and output neuron spikes. To test the correctness of our method, we have performed a battery of simulations as follows: (i) for all simulations an input neuron spike V_{pre} occurs at $100ns$ and ends at $1.1\mu s$ (ii) for each simulation an output neuron spike V_{post} occurs with a certain delay after the input neuron spike. For each simulation a different delay is considered such that all cases of potentiation and depression can be observed. Since we deal with probabilistic switching, each simulation was repeated 20 times to account for stochastic effects. The results are illustrated in Fig. 3d, where the synaptic conductance is plotted as a function of the delay between post and pre-synaptic signal. On the x-axis we mark the delay of the post-synaptic signal compared to the pre-synaptic one, while on the y-axis we mark the equivalent conductance of the synapse (obtained by averaging the values obtained for the 20 simulations performed under identical conditions). The delay-dependent potentiation and depression are clearly demonstrated.

V. Results & Discussion

In this section, we explain how the values of the control signals have been chosen for the analysis in Section IV. We first studied the behaviour of the synapse under a constant voltage drop to determine the range of values for V_{prob} defined in Section IV, (i.e., the voltage amplitude for which the 4 MTJs will switch at different times due to the stochasticity of the switching process). The synapse designed with 4 MTJs in parallel has five conductance states: from state0 (maximum conductance) to state4 (minimum conductance). To analyse the synaptic potentiation, the synapse is initialized at state4 (the bottom left corner in Fig. 4a) and different voltage signals applied to it (varying both their amplitude and duration). The colormap in Fig. 4a) displays the average conductance over 10 simulations for each applied voltage signal (couple pulse amplitude, pulse duration). It should be noted that for voltage amplitudes of around $340mV$ all 5 conductance levels can be achieved for different pulse widths, while for voltage amplitude of $360mV$ and pulse width larger than $40ns$ there is a sudden change in conductance from its minimum to its maximum value. The synaptic depression is analysed in a similar way and it is illustrated in Fig. 4b). The results illustrated by the 2 colormaps offer the complete information on the Synaptic behavior and allow choosing the shape and parameter values for the pre- and post- synaptic signals described in section IV. For instance, the $50ns$ width of the positive and negative parts of the postsynaptic pulse V_{post} was chosen because the analysis shows that for both potentiation and depression, $50ns$ pulse width allows the synapse to take all the possible synaptic states depending on the voltage amplitude. Simulation parameters are summrised in Table I.

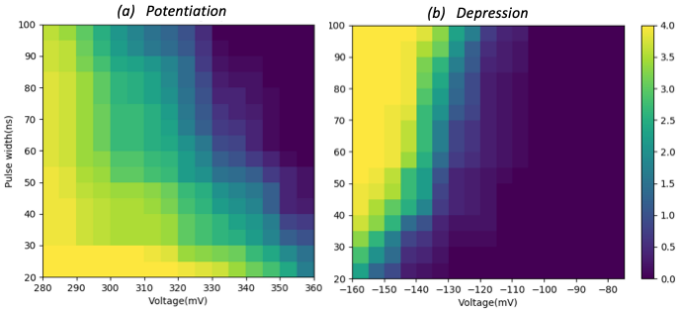

Fig. 4. (a)Potentiation: The state of the 4 MTJs synapse is initialized at state4, a positive square pulse voltage with different widths is applied to lower the resistance of the synapse. Each point is the average of 10 simulations. (b) Depression: Negative square voltages of different widths are applied to a synapse initialized at state0 (bottom right corner)

TABLE I
Simulation Parameters

	Description	Value
Input spike	Max voltage at the beginning	250mV
	Voltage at the end	-60mV
	Pulse width	$1.1\mu s$
	Starting delay	100ns
Output spike	Voltage at the positive part	100mV
	Width at the positive part	50ns
	Voltage at the negative part	-100mV
	Width at the negative part	50ns
Magnetic Tunnel Juncion MTJ	Distribution of thermal fluctuation	exponential
	Oxide thickness	8.5 Å
	Magnetic free layer thickness	1.3nm
	total thickness of MTJ nanopilar	33.55 nm

Fig 5 shows the increase and decrease in synapse conductance ΔG, this depends only on the relative timing Δt between presynaptic and postsynaptic spikes. The synapse is initialized at an intermediate state, where two MTJs are in a high resistance state and two are in a low resistance state. The parameters giving the profiles of the input and output spikes are given in Table I. Figure 6 shows the behavior of the synapse when initialised at the lowest (left) and highest (wright) conductance respectively. Figure 7 shows the effect of the slope of the pre-synaptic signal (in Fig. 3) on the potentiation and depression efficiency. We found that a best

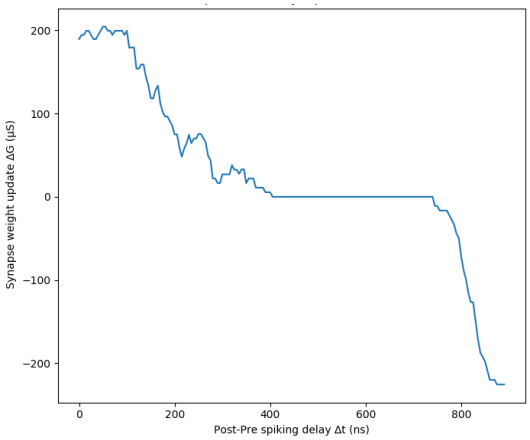

Fig. 5. Plasticity of the synapse. The conductance change ΔG in the synapse with respect to the relative timing Δt between input and output spikes. For each Δt, we show the average ΔG over 20 simulations

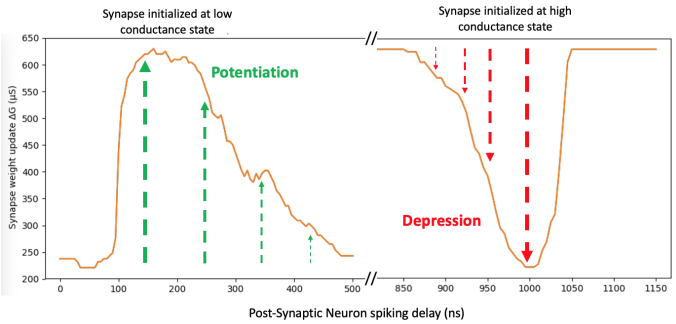

Fig. 6. The plasticity of a 4 MTJ synapse. Initialized at the lowest then at the highest conductance respectively

and more resolved potentiation and depression are obtained for larger signal slopes (in absolute value).

VI. Conclusion

Spiking Neural Networks are attractive solutions for artificial intelligence applications, their implementation in hardware is very energy efficient. We presented in this paper a simple way to implement the synaptic connections using the magnetic tunnel junctions, we seceded to reproduce the most important parameter for the STDP training rule which is time-dependence. The training which consists of adjusting the synapse conductance is only controlled by the time delay between the spikes of the connected neurons to the synapse. This way the synapse is designed to be used in a bigger network to allow unsupervised learning. The voltage pulse shapes of input and output neurons are chosen in such a way to allow the occurrence of the desired operation (potentiation or depression) and its magnitude at the right time. Moreover, potentiation and depression occur only when the two incoming pulses arrive, making the design suitable also for inference, where the synapse conductance stays unchanged when only one spike arrives in the synapse.

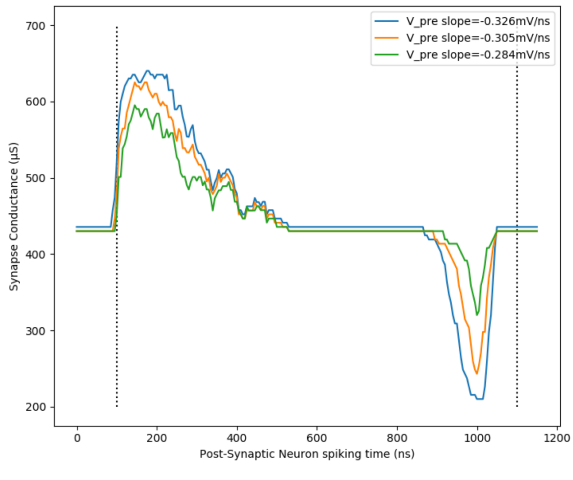

Fig. 7. Potentiation and depression of a synapse initialized at an intermediate state of conductance, the presynaptic pulse is fixed at 100ns and is delimited by the two vertical line. Three simulations with different pulses are shown. a better result is obtained for slope of $-0.326mV/ns$

References

[1] LeCun, Y., et al (1989). Handwritten digit recognition with a back-propagation network. Adv. in neural information processing systems.

[2] Ballard, Z., Brown, C., Madni, A. M., Ozcan, A. (2021). Machine learning and computation-enabled intelligent sensor design. Nature Machine Intelligence, 3(7), 556-565.

[3] Marković, D., Mizrahi, A., Querlioz, D., Grollier, J. (2020). Physics for neuromorphic computing. Nature Reviews Physics, 2(9), 499-510.

[4] Strong, S. P., et al. (1998). Entropy and information in neural spike trains. Physical review letters, 80(1), 197.

[5] Kok, M. R. (2020). Biologically Realistic Artificial Neural Networks (Bachelor's thesis, University of Twente).

[6] Y. Zhang et al., Compact Model of Subvolume MTJ and Its Design Application at Nanoscale Technology Nodes, IEEE Transactions on Electron Device, 2015, 62:2048-2055. DOI: 10.1109/TED.2015.2414721

[7] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830.

[8] Pfeiffer, M. and Pfeil, T., 2018. Deep Learning With Spiking Neurons: Opportunities and Challenges. Frontiers in Neuroscience, 12.

[9] Merolla, P. A., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 345, 668–673. doi: 10.1126/science.1254642

[10] Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2013). Overview of the SpiNNaker system architecture. IEEE Trans. Comput. 62, 2454–2467. doi: 10.1109/TC.2012.142

[11] D. Querlioz, O. Bichler, P. Dollfus and C. Gamrat, "Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices," in IEEE Transactions on Nanotechnology, vol. 12, no. 3, pp. 288-295, May 2013, doi: 10.1109/TNANO.2013.2250995.

[12] Zhang, D., Zhao, W., Zeng, L., Cao, K., Wang, M., Peng, S., . . . Wang, Y. (2016). All Spin Artificial Neural Networks Based on Compound Spintronic Synapse and Neuron. IEEE Transactions on Biomedical Circuits and Systems, 10(4), 828-836. doi:10.1109/tbcas.2016.2533798

[13] Diehl, P. U., Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in computational neuroscience, 9, 99.

[14] Maranhão, G., Guimarães, J. G. (2021). Low-power hybrid memristor-CMOS spiking neuromorphic STDP learning system. IET Circuits, Devices Systems, 15(3), 237-250.

[15] Andreeva, N. V., Ryndin, E. A., Gerasimova, M. I. (2020). Memristive logic design of multifunctional spiking neural network with unsupervised learning. BioNanoScience, 10(4), 824-833.