

A Stochastic Approach to STDP

Runchun Wang, Chetan Singh Thakur, Tara Julia Hamilton, Jonathan Tapson, André van Schaik
The MARCS Institute, Western Sydney University, Sydney, NSW, Australia
mark.wang@westernsydney.edu.au

Abstract— We present a digital implementation of the Spike Timing Dependent Plasticity (STDP) learning rule. The proposed digital implementation consists of an exponential decay (exp-decay) generator array and a STDP adaptor array. The weight values are stored in a digital memory, and the STDP adaptor will send these values to the exp-decay generator using a digital spike of which the duration is modulated according to these values. The exp-decay generator will then generate an exponential decay, which will be used by the STDP adaptor for performing the weight adaption. The exponential decay, which is computationally expensive, is efficiently implemented by using a novel stochastic approach. This stochastic approach was fully analysed and characterised. We use a time multiplexing approach to achieve 8192 (8k) virtual STDP adaptors and exp-decay generators with only one physical adaptor and exp-decay generator respectively. We have validated our stochastic STDP approach with measurement results of a balanced excitation experiment. In that experiment, the competition (induced by STDP) between the synapses can establish a bimodal distribution of the synaptic weights: either towards zero (*weak*) or the maximum (*strong*) values. Our stochastic approach is therefore ideal for implementing the STDP learning rule in large-scale spiking neural networks running in real time.

I. BACKGROUND

The Spike Timing Dependent Plasticity (STDP) algorithm [1], which has been observed in the mammalian brain, modulates the weight of a synapse based on the relative timing of pre-synaptic and post-synaptic spikes. In STDP, the synaptic weight will be increased (or decreased) if a pre-synaptic spike arrives several milliseconds before (or after) the post-synaptic spike fires. This learning rule is computationally intensive as it requires a lot of exponential and division functions.

In neuromorphic systems, various implementations of the STDP algorithm are proposed such as, a circuit based on analogue blocks and flip-flops [2], a bistable synapse with a very compact analogue implementation of STDP [3], analogue blocks and switches to implement exponential STDP [4], and a digital synapse with an adaptive kernel, binary update rule and shift-based homeostasis [5]. We have previously presented a compact implementation of the STDP using linear decays [6], [7]. Here, we present its follow-up work that uses a novel stochastic approach that can efficiently implement the

This work has been supported by the Australian Research Council Grant DP140103001. This work was inspired by the Capo Caccia Cognitive Neuromorphic Engineering Workshop 2014 and Telluride Neuromorphic workshop 2015.

exponential-type STDP, inspired by our recent work on stochastic electronics as a novel way of building circuits [8].

II. STOCHASTIC DECAY

A. Infinite Impulse Response (IIR) filter approach

A discrete time first order IIR filter can be expressed by the following equation:

$$V[t + 1] = \alpha V[t] \quad (1)$$

where, t represents the index of the time step, and $V[t]$ represent the previous value of V and the IIR filter constant α is defined as:

$$\alpha = \frac{\tau}{\tau + 1} \quad (2)$$

where, τ is the time constant and the decay d is given by:

$$d = V[t] - V[t + 1] = \frac{V[t]}{\tau + 1} \quad (3)$$

When τ is large, α is only a little less than 1, and a large number of bits are needed to encode its value accurately. If the number of bits used to encode V is equal to, or less than, the number of bits used to encode α , the above recursive multiplication just results in a near linear decay.

This situation occurs, for example, when simulating a neural network with many millions of neurons using the time multiplexing (TM) approach [9–11]. With a standard IIR filter approach, a large number of bits would be needed for each state variable to calculate long time constants. In addition, large memory storage per state variable will result in a communication bottleneck, since only a few bits can be exchanged with the memory in a single clock cycle.

B. Stochastic approach

We have used a stochastic approach to implement long time constants in hardware using fewer numbers of bits for the state variables in our previous work [12]. The work reported here was based on that work. In this implementation, we first multiply V by the IIR factor α and then add a random number r to the multiplication result. Mathematically, the method can be written as:

$$V[t + 1] = \text{int}\left(\frac{\tau}{\tau + 1}V[t] + r[t]\right) \quad (4)$$

where, $r[t]$ is a random number drawn from a uniform distribution in the range (0,1). This is effectively a form of dithering to deal with the rounding of V , which is a 4-bit number, to an integer value. The probability of decaying p is then given by:

$$p = \left(r < \frac{V}{\tau + 1} \right) = \frac{V}{\tau + 1} \quad (5)$$

The decay is then given by:

$$d = V[t] - V[t + 1] \quad (6)$$

$$= \text{int} \left(\frac{V[t]}{\tau + 1} - r[t] \right) \quad (7)$$

$$= \text{int} \left(\frac{V}{\tau + 1} \right) + X \quad (8)$$

$$P(X = 1) = \frac{V[t]}{\tau + 1} \% 1 \quad (9)$$

where, $x \% 1$ is x modulo 1 and $\text{int} (V/(\tau + 1))$ is the integer part of $V/(\tau + 1)$. This shows immediate parallels with the original IIR filter and our stochastic approach is capable of producing the same exponential decay. We only store few MSBs of the final product, e.g., V is stored as a 4-bit integer.

C. Characterisation of variances

This stochastic approach not only reduces the storage needed, but also introduces variability between the STDP synapses while using the exact same synapse model. This makes the networks more realistic simulations of biological neural networks. The variance for a single decay is given by:

$$\text{Var}(n) = \frac{1 - p}{p^2} = \frac{(\tau + 1)^2}{V^2} - \frac{\tau + 1}{V} \quad (10)$$

Since each duration until a decrement is an independent random variable, the variance for the half-time h is given by the sum of the variances for each decrement:

$$\text{Var}(h) = \sum_{i=1}^{V/2} \frac{(\tau + 1)^2}{(V + 1 - i)^2} - \frac{\tau + 1}{V + 1 - i} \quad (11)$$

$$= \sum_{i=1}^{V/2} \frac{(\tau + 1)^2}{(V/2 + i)^2} - \sum_{i=1}^{V/2} \frac{\tau + 1}{V/2 + i} \quad (12)$$

In this equation, the sum converges to $\ln(2)$ as $V/2 \rightarrow \infty$, so that we can write:

$$\text{Var}(h) \approx \frac{(\tau + 1)^2}{V + 1} - (\tau + 1) \ln(2) \quad (13)$$

The variances of the half-time h will be very large when τ is large. Mathematically, the larger range r is in, the bigger the variances for the half-time h will be. In the above analysis, r is from a uniform distribution in the range (0,1). Reducing the variances for the half-time h can be effectively achieved by limiting r in a smaller range as long as the following condition is met:

$$\alpha V + \min(r) < V \quad (14)$$

Otherwise V will not decay. It is obvious that the most critical condition is when V is 1. Since for digital implementations, the most efficient way to generate random numbers is to use linear feedback shift registers (LFSRs), this condition can be expressed as:

$$\frac{\tau}{\tau + 1} + \min(r) = \frac{\tau}{\tau + 1} + \frac{1}{2^L} < 1 \quad (15)$$

$$\tau < 2^L - 1 \quad (16)$$

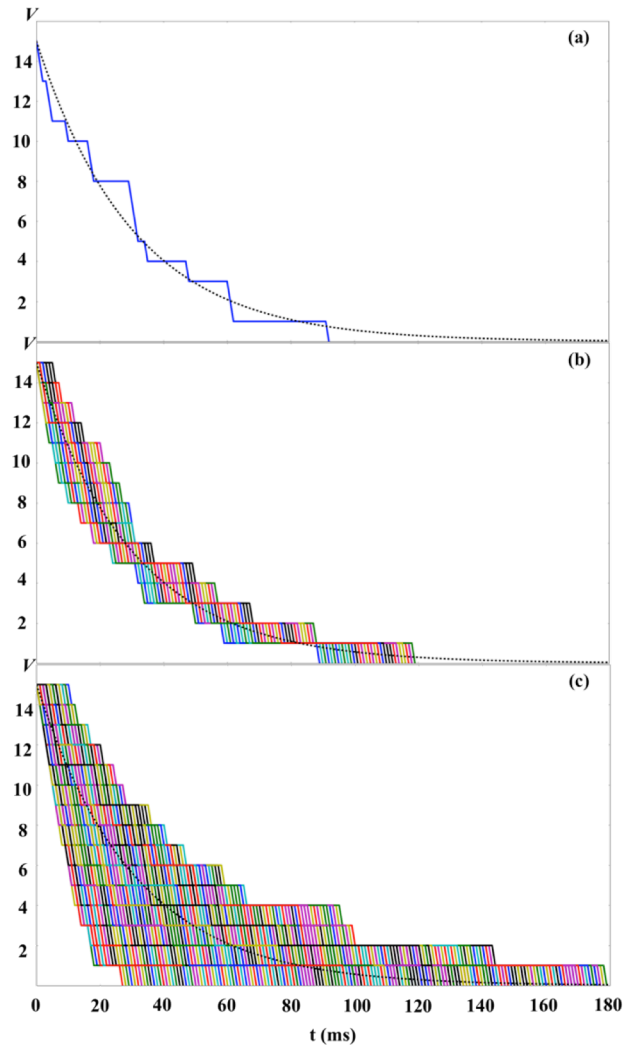


Fig. 1. Exponential decay obtained by using the stochastic approach. The dashed line is the IIR decay trace with a time constant τ of 30 ms ($\alpha = 495/512$, a 9-bit number). V is stored as a 4-bit integer with an initial value of 15. (a-b) One exponential decay and all the exponential decays achieved by a 5-bit LFSR respectively; and (c) Exponential decays achieved by a 9-bit LFSR, using 1023 different random seeds. It is clear that the variances of the exponential decays achieved by the 9-bit LFSR are much larger than the ones of the decays achieved by the 5-bit LFSR.

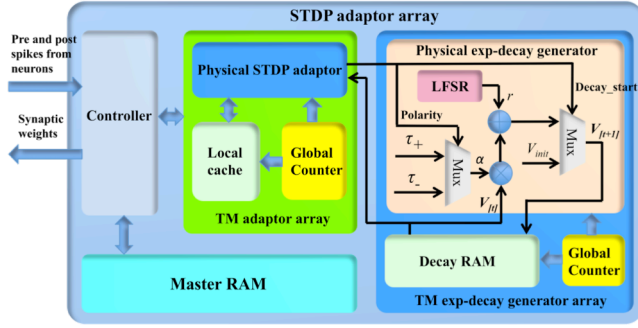


Fig. 2. The structure of the STDP adaptor array.

where, L is the length of the LFSR. For example, the maximum time constant that a 5-bit LFSR can achieve is 30 ms (the time step is 1 ms). Using a 9-bit LFSR for the same time constant will create much larger variances (see Fig. 1). Hence the principle to reduce to the variances of the half-time h is to use the LFSR with the minimum length that can still achieve the time constant.

III. HARDWARE IMPLEMENTATION

A. Learning rule

In our hardware implementation, the amount of synaptic modification is summarised by the following equations:

$$\Delta w = \begin{cases} A^+ \exp(\Delta t / \tau_+), & \text{if } \Delta t < 0 \\ -A^- \exp(\Delta t / \tau_-), & \text{if } \Delta t \geq 0 \end{cases} \quad (20)$$

where, Δw is the modification of the synaptic weight, Δt is the arrival time difference between the pre- and post-synaptic spike. A^+ and A^- determine the maximum amounts of synaptic modification for each spike pair. The τ_+ and τ_- are time constants and control the rate of decay for potentiation and depression portions of the curve. As we focus on the low-cost hardware implementation of the exponential-type STDP, quantifying the effects of our learning rules on the synaptic weight are out of the scope of this paper. In the work reported here, we use $\tau_+ = \tau_- = 20$ ms and $A^+ = A^- = 1$ throughout. Hence, the Δw is indeed the $V_{[t]}$ in equation (4).

B. Topology

In our previous work [6, 7], we implemented a synaptic plasticity adaptor array that is separate from the neurons in the neural network. Each adaptor (in that array) performs synaptic plasticity, i.e., STDP, according to the arrival times of the pre- and post-synaptic spikes assigned to it. And it sends out the updated result, i.e., synaptic weight, to the post-synaptic neuron in the neural network. Since this strategy provides great flexibility for building complex large-scale neural networks, we chose to use exact the same architecture as in [6] to implement an exponential-type STDP adaptor array (see Fig. 2). It consists of a controller, a Master RAM, a TM STDP adaptor array and a TM exp-decay generator array, all of which, with the exception of the last one, are identical to ones

presented in [6]. The TM adaptor array and the TM exp-decay generator array are both configured to have 8192 (8k) units, each TM exp-decay generator being assigned to one TM STDP adaptor. The TM time window generator array in [6], which generates linear decay, is replaced by the exp-decay generator array.

The exponential-type STDP adaptor array operates in the exact same manner as the digital synaptic adaptor array in [6]. The controller receives pre- and post-synaptic spikes from the neuron array and assigns them to the corresponding TM STDP adaptors according to their addresses. The weight values are stored in the local cache and the Master RAM. The TM STDP adaptors will send these values to the TM exp-decay generator array using a digital spike of which the duration is modulated according to these values.

Each TM exp-decay generator will start an exponential decay when either a pre- or post-synaptic spike arrives, which will be used by the corresponding TM STDP adaptor to determine the weight update. As we assume that the adaption will not be carried out if the pre- and post-synaptic spikes arrive simultaneously, thus only one TM exp-decay generator will be needed. The STDP adaptor will carry out the weight adaption using its output $V_{[t]}$. The stored weight values will also be sent out to the corresponding neuron in the neural network for the post-synaptic current generation.

C. TM exp-decay generator array

The TM exp-decay generator array was implemented using the TM approach [9–11] to achieve 8k TM exp-decay generators with only one physical exp-decay generator. The global counter processes each TM exp-decay generator sequentially. Each TM exp-decay generator uses a time slot of 25 clock cycles (125 ns with 200 MHz clock frequency) to complete its processing to maintain an update rate of 1 kHz (the corresponding time step is about 125 ns × 8k = 1 ms).

In each time slot, the global counter will read the value of the $V_{[t]}$ (a 4-bit integer) from the Decay RAM with a size of 8k × 4bit. $V_{[t]}$ will be reset to V_{init} (set to 15 here), when the digital input spike (Decay_start) from the TM adaptor is active (high). When there is no input spike, we will apply the stochastic approach (see equation (4)) to $V_{[t]}$ on each time slot (of that TM exp-decay generator), until it reaches zero, indicating the end of the exponential decay.

These computations was implemented with a single fixed-point number multiplier. Its inputs are α (a 9-bit integer) and $V_{[t]}$ (4-bit) and its output result is 13-bit wide. For future extension capability, we used a multiplexer to choose α for different time constants. For the same reason, we used a 7-bit LFSR to generate r . The LFSR is configured to use its five least significant bits (LSBs) in the work reported here ($\tau_+ = \tau_- = 20$ ms) and it will generate a new value every 1 ms. The product $V_{[t+1]}$ will then be stored into the exp-decay RAM.

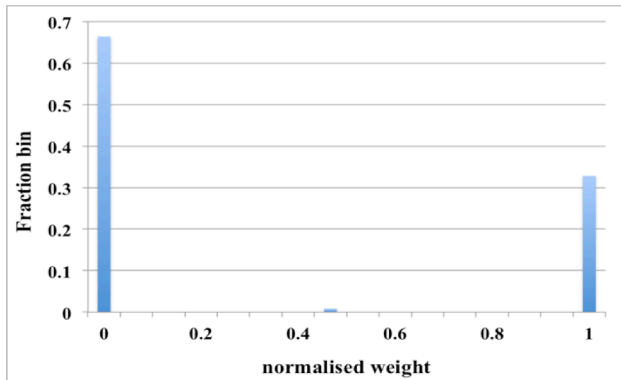


Fig. 3. Balanced excitation experiment. (a) Weight distribution after 1.25s of STDP for an input rate of 10 Hz. The bimodal distribution of *strong* and *weak* weights is apparent.

IV. MEASUREMENT RESULTS

We have successfully implemented the exponential-type STDP adaptor array on an Altera Cyclone V FPGA. Table I shows the utilisation (without the Master RAM) of hardware resources on the FPGA. As Table I shows, the proposed system uses only a small fraction (<1%) of the hardware resources.

We have tested the performance of the exponential-type STDP adaptor array by performing a balanced excitation experiment, based on the experiment run by [13]. Song et al. (2000) have shown that competitive Hebbian learning [14] can be performed through STDP. The competition (induced by STDP) between the synapses can establish a bimodal distribution of the synaptic weights: either towards zero (*weak*) or the maximum (*strong*) values (see Fig. 3).

V. CONCLUSIONS

In this paper, we demonstrated a digital implementation of the STDP learning rule using a novel stochastic approach. This approach is capable of producing the same results to a more complex STDP model while occupying only a fraction of the area. The compactness plus the variability presents a perfect solution for implementing synaptic learning in large-scale digital neural networks.

VI. REFERENCES

[1] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, no. 6595, pp. 76–81, Sep. 1996.

[2] A. Bofill-i-petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1296–304, Sep. 2004.

[3] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses With Spike-Timing Dependent Plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, Jan. 2006.

TABLE I

Device utilisation Altera Cyclone 5CGXFC5C6F27C7

Layers	ALMs	RAMs	DSPs
1	246/29080	192k/4.5M	1/450

[4] T. J. Koickal, A. Hamilton, S. L. Tan, J. A. Covington, J. W. Gardner, and T. C. Pearce, "Analog VLSI Circuit Implementation of an Adaptive Neuromorphic Olfaction Chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 60–73, Jan. 2007.

[5] S. Afshar, L. George, C. S. Thakur, J. Tapson, A. van Schaik, P. de Chazal, and T. J. Hamilton, "Turn Down That Noise: Synaptic Encoding of Afferent SNR in a Single Spiking Neuron," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 2, pp. 188–96, 2015.

[6] R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A compact reconfigurable mixed-signal implementation of synaptic plasticity in spiking neurons," in *IEEE International Symposium on Circuits and Systems*, 2014, pp. 862–865.

[7] R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A neuromorphic implementation of multiple spike-timing synaptic plasticity rules for large-scale neural networks," *Frontiers in Neuroscience*, 2015.

[8] T. J. Hamilton, S. Afshar, A. van Schaik, and J. Tapson, "Stochastic Electronics: a neuro-inspired design paradigm for integrated circuits," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 843–859, May 2014.

[9] R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A mixed-signal implementation of a polychronous spiking neural network with delay adaptation," *Frontiers in Neuroscience*, vol. 8, no. March, pp. 1–16, Mar. 2014.

[10] R. M. Wang, G. K. Cohen, K. M. Stiefel, T. J. Hamilton, J. Tapson, and A. van Schaik, "An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation," *Frontiers in Neuroscience*, vol. 7, no. February, pp. 1–14, 2013.

[11] R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A compact neural core for digital implementation of the Neural Engineering Framework," in *BIOCAS2014*, 2014, pp. 548–551.

[12] R. M. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "An FPGA design framework for large-scale spiking neural networks," *IEEE International Symposium on Circuits and Systems*, 2014, pp. 457–460.

[13] L. F. Abbott, S. Song, and K. D. Miller, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, Sep. 2000.

[14] D. O. Hebb, *The organization of behavior*. New York, NY: Wiley & Sons, 1949.