Error-feedback-based auto-tuning algorithm for faster memristor programming

Sushmit Hossain, Buyun Chen, Zerui Liu, Yunxiang Wang, Ryan Bena, Hsiang-Chun Cheng, Quan Nguyen, Mike Shuo-Wei Chen and Wei Wu*
Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, United States
**wu.w@usc.edu*

In recent years, memristors have been used in various applications such as-hardware-based neural networks [1], Analog Computing [2], Robotics [2], etc. The computing applications and the backpropagation algorithm used in the neural network training phase require memristors to be tuned to specific conductance levels in a very short time. The algorithm mostly used in programming the memristor is the "write-verify algorithm" [3]. In this work, we present a modified version of this algorithm where we use the error in conductance value as feedback for the next tuning cycle. We also use a feedback-based reset method to keep the memristor in the "soft-reset" regime. The results show significant improvement over the traditional method.

Figure 1a shows the multiple tunable states of our $Pt/Ta/Al_2O_3/Pt$ memristors. Each current pulse applied sets the memristor to a specific conductance (Figure 1b). For this demonstration, we tuned the memristors to the range 700Ω-3500Ω, in this range our tuning algorithm can achieve any resistance value within 20Ω tolerance.

Figure 2 shows our tuning algorithm. The initial current applied to the memristor can be found from the fitting function of Figure-1b. After each current pulse, the conductance value will be checked and error between the target and achieved conductance will be calculated. After multiplying a proportional gain "K" by this error value, the next current pulse amplitude can be calculated. If at some point, the conductance goes higher than the target value, a soft reset will be performed with the resistance limited to the target value. This iterative process will continue until a certain resistance is reached within the preferred tolerance range.

We have tested this tuning algorithm in our FPGA-based ultra-fast tuning setup. 10ns current pulses were used as SET pulses in this setup (Figure 3a). With careful optimization of the proportional gain, K, the memristor can be tuned to any value within 700-3.5k within the average 15 set-reset pulses (Figure 3b), whereas the traditional "write-verify" approach takes around 100 pulses to tune memristors to certain conductance with such accuracy. Additionally, with the help of this algorithm, a higher number of states can be achieved than reported before, as long as there is no strict upper limit to the number cycles required for tuning.

---

[1] Yao, P., Wu, H., Gao, B. et al. Fully hardware-implemented memristor convolutional neural network. Nature 577, 641–646 (2020).

[2] Chen B, Yang H, Song B, Meng D, Yan X, Li Y, Wang Y, Hu P, Ou TH, Barnell M, Wu Q, Wang H, Wu W. A memristor-based hybrid analog-digital computing platform for mobile robotics. Sci Robot. 2020 Oct 21

[3] Zidan, M.A., Jeong, Y., Lee, J. et al. A general memristor-based partial differential equation solver. Nat Electron 1, 411–420 (2018).

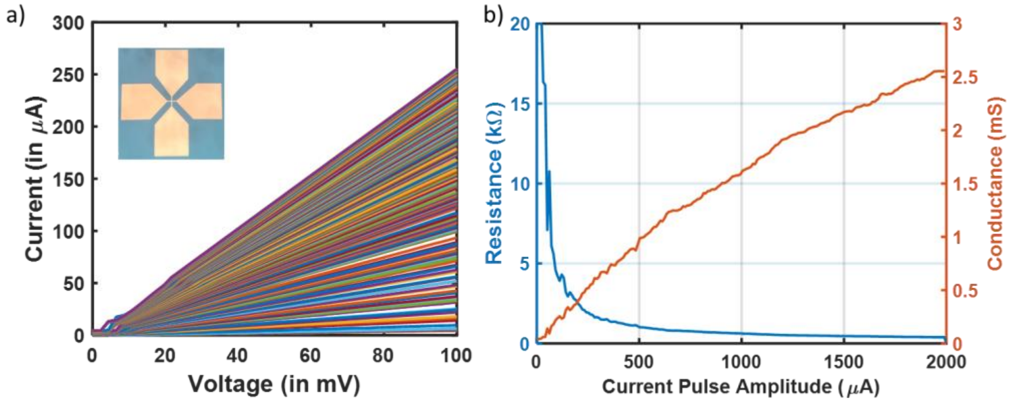

Figure 1: a) Multiple states of our memristor. b) Resistance and conductance values of the memristor after each current pulse.

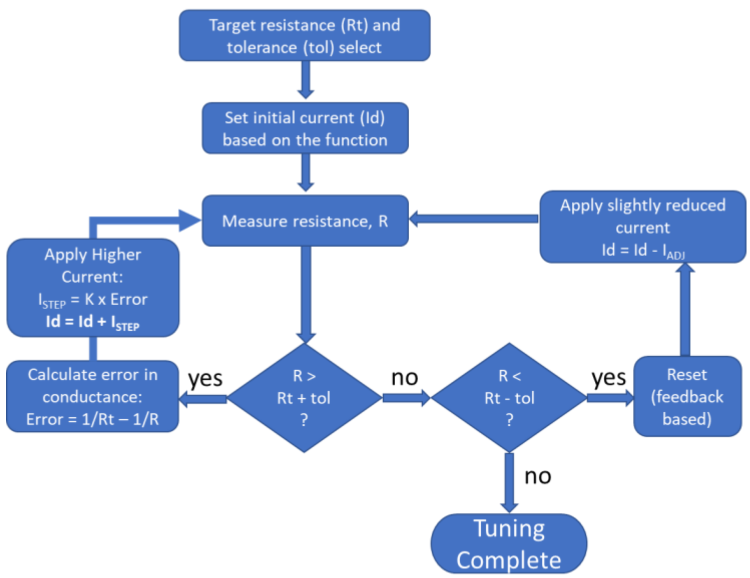

Figure 2: Error-feedback based auto-tuning algorithm.

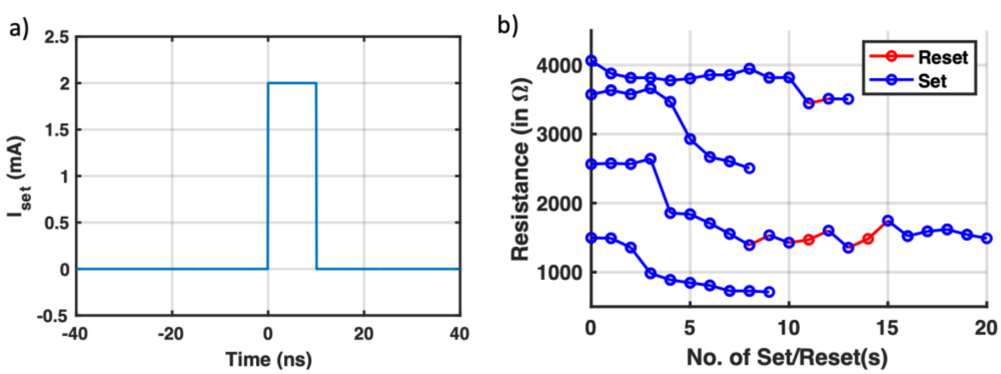

Figure 3: a) 10ns SET pulse. The amplitude ranged from 50μA to 2mA. b) Experimental results of the tuning process. Memristors were tuned to four different resistance values – 700 Ω, 1500 Ω, 2500 Ω and 3500 Ω with 20 Ω tolerance limit.