

CMOS-based Stochastically Spiking Neural Network for Optimization under Uncertainties

Amit Ranjan Trivedi¹ and Saibal Mukhopadhyay²

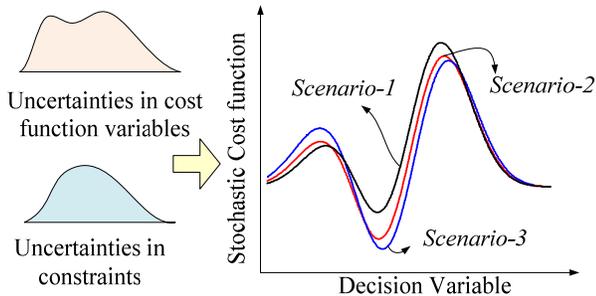
¹Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607
¹Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Abstract: We present CMOS-based ‘stochastically spiking neural network’ for optimization under uncertainties. We discuss a ‘scenario generation’ circuit to non-parametrically estimate/emulate statistics of uncertain cost/constraints variables in an optimization problem. We also present a ‘spiking neural network’ for linear/quadratic programming. Scenario generation block stochastically controls spiking neural network to extract optimal solution of an optimization problem minimizing its expected cost.

Keywords: Spiking Neural Network; Optimization; Stochastic Programming.

Introduction

This work presents CMOS-based ‘stochastically spiking neural network’ for optimization under uncertainties. In several real-world problems (such as demand-supply and networking), constraints and/or cost functions of the optimization problem are uncertain and only statistically defined [1] [Fig. 1]. Stochastic programming algorithms (such as sample average approximations [2], stochastic decomposition [3]) have been developed to extract optimal variables minimizing expected value of cost under such uncertainties. However, a purely algorithmic implementation of stochastic programming is inefficient in power and performance, and especially for large scale problems. This work presents a stochastically spiking neural network for a large scale, high performance optimization under uncertainties. The following key novelties are explored: (1) We discuss a ‘scenario generation’ circuit to non-parametrically estimate and



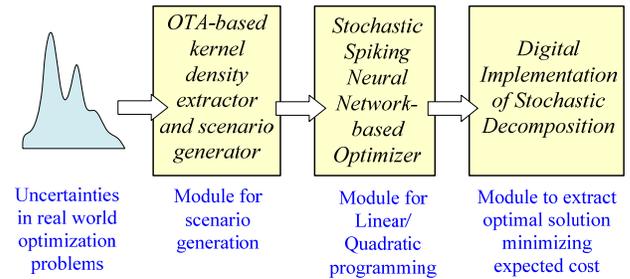
Stochastic Programming: Optimization under uncertainties

Figure 1: Cost function scenarios under uncertain and statistically defined cost function variables and/or constraints.

emulate statistics of uncertain cost/constraints variables. (2) We present a ‘spiking neural network’ for linear and quadratic programming. Scenario generation block and spiking neural network are integrated for a stochastically spiking neural network which extracts optimal variables minimizing expected cost in an optimization problem with uncertainties. The discussed mixed-signal, CMOS-based architecture of stochastically spiking neural network minimizes area/power of each cell and enables a large scale integration [Fig. 2].

Scenario Generation Module

Kernel density method can non-parametrically estimate density of a random variable based on its observed



System Diagram for Stochastic Programming

Figure 2: System diagram integrating scenario generation units, spiking neural network, and stochastic decomposition.

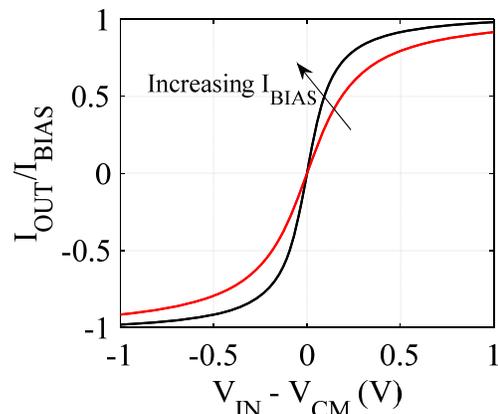


Figure 3: $I_{OUT} - V_{IN}$ of Operational Transconductance Amplifier (OTA) is utilized for Kernel function estimation.

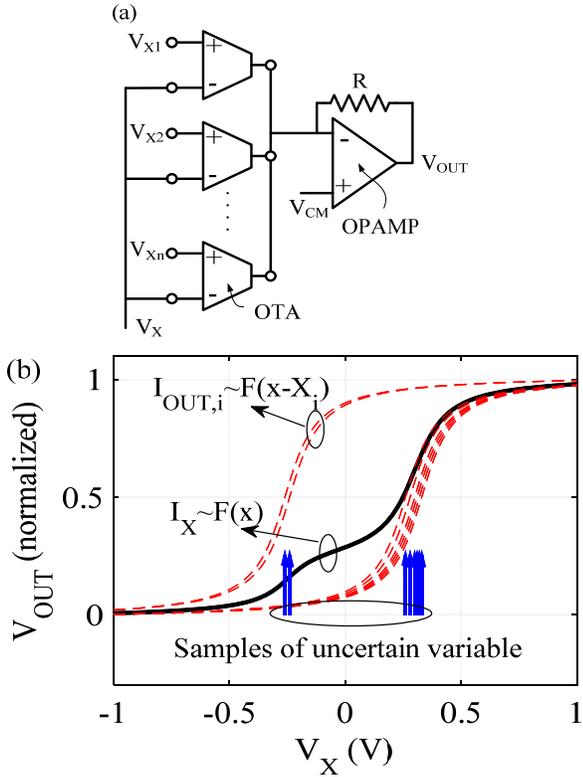


Figure 4: OTA column (a) adds current from each cell to estimate CDF at x in (b).

samples. Cumulative density function (CDF), $F(x)$, of a random variable (x) is expressed as $F(x) = \frac{1}{N} \sum_{i=1}^N G\left(\frac{x-X_i}{h}\right)$. Here, X_i are the observed samples of x , and N are the total number of samples. $G(x)$ is the Kernel function for CDF estimation having a local density, and h is the Kernel function width. We utilize an operational transconductance amplifier (OTA) [4] to physically implement a Kernel function. Output current, I_{OUT} , of an OTA follows inverse tangent characteristics at varying input voltage (V_{IN}) [Fig. 3], thereby it is suitable for Kernel function implementation. By varying bias current of OTA, $I_{OUT}-V_{IN}$ characteristics can be modulated to realize varying Kernel function width. Fig. 4 shows CDF estimation through an OTA column. OTAs in the column are connected in parallel. One of the input terminal of OTAs receives test voltage, V_x , and the other observed sample voltage, X_i . An OP-AMP at the output of the column integrates currents from various OTAs. Output of the OP-AMP represents CDF value at V_x . Scenarios on cost function/constraint variables are generated based on inverse transform on CDF. In Fig. 5, $F^{-1}(u)$ for uniformly distributed random number $u \in [0, 1]$ extracts random samples of x varying with CDF of $F(x)$. In Fig. 6, we present a successive approximation (SA) circuit to evaluate inverse transform on CDF. A uniform random number

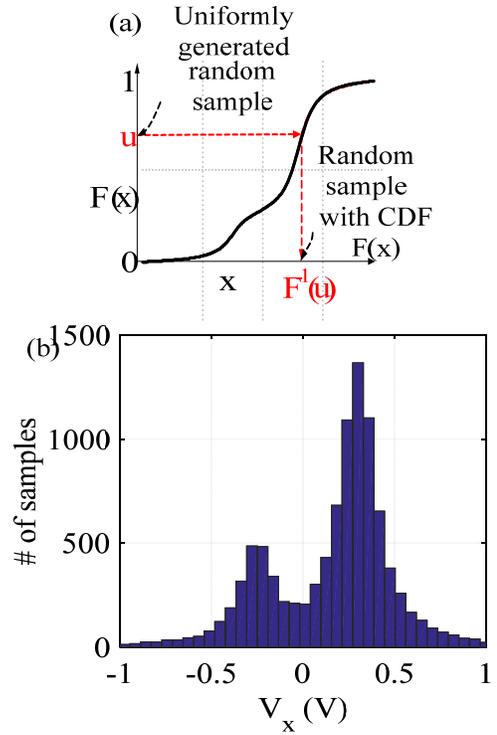


Figure 5: (a) Inverse transform on CDF to extract random sample of variable x . (b) Histogram of samples.

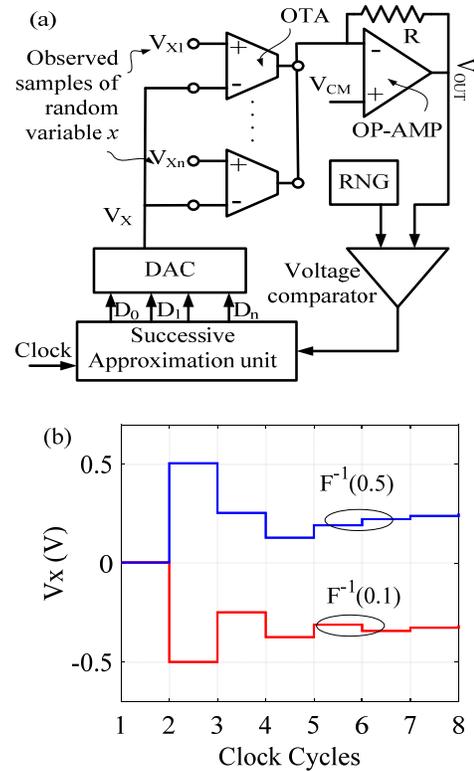


Figure 6: (a) Successive approximation circuit for inverse transform evaluation on CDF. (b) Inverse transform transients.

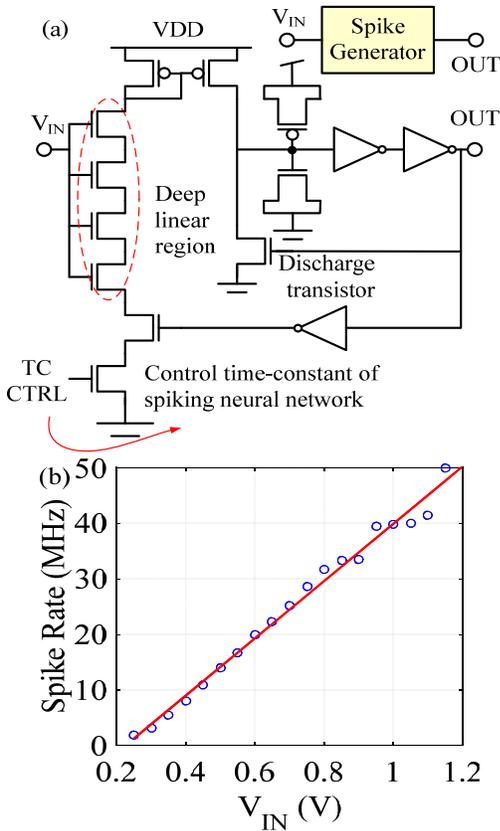


Figure 7: OTA column (a) adds current from each cell to estimate CDF at x in (b).

generator (RNG) generates a sample value u . SA circuit evaluates $F(x_{in})$ with an initial estimate x_{in} . The estimate is updated in each clock cycle by comparing $F(x_{in})$ with u . Most significant to least significant bits in the digital representation of x_{in} are updated in each clock cycle for $F(x_{in}) = u$ in Fig. 6(b). Thereby, with a random uniform distribution of u , random samples of x are extracted following $CDF = F(x)$.

Stochastically Spiking Neural Network for Optimization

We explore spiking neural network implementation of linear and quadratic programming algorithms presented in [5]. Fig. 7(a) shows a spike generation circuit with linearly increasing spike rate at increasing input voltage. The circuit utilizes a chain of series connected NMOS to realize linearly voltage-controlled current by pushing the transistors in the chain in a deep linear region. In Fig. 7(b), Spike rate linearly increases with increasing input voltage, V_{IN} , beyond a cut-off of $\sim 0.2V$. Voltage at ‘TC CTRL’ in Fig. 7(a) proportionally increases or reduces spike rate at a V_{IN} to program time-constant of the network, and enables power-performance trade-off. Circuit in Fig. 8(a) is triggered by a spike, and produces a pulse controlled by a digital weight-bits D [transient waveforms in Fig. 8(b)]. The circuit in Fig. 9(a) receives input pulses from the pulse

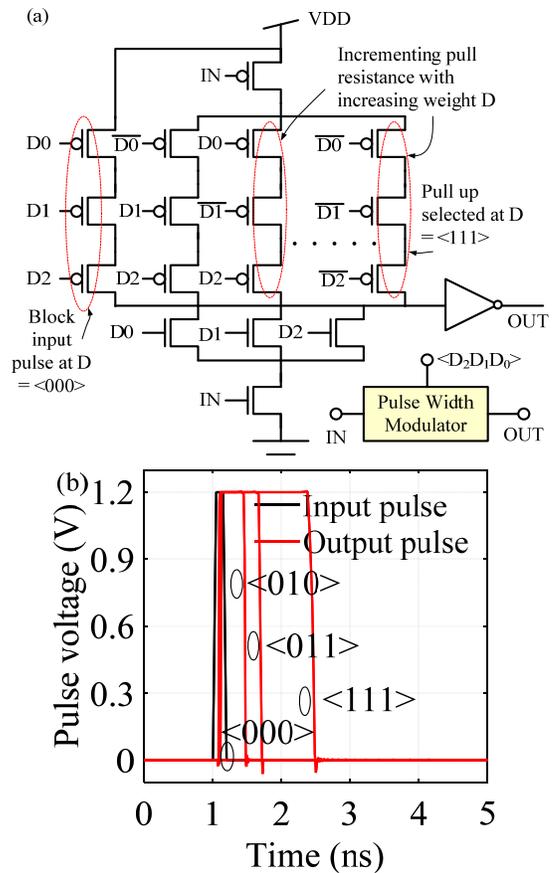


Figure 8: (a) Pulse width modulation circuit. (b) Pulse width control at varying digital weight bits $\langle D_2D_1D_0 \rangle$.

modulator. Depending on the weight sign bit, these pulses charge or discharge the membrane capacitance to generate an analog output. This analog output is applied to the spike generator block to produce corresponding spike frequency. Another critical unit for the linear/quadratic programming algorithm in [5] is a derivative sign unit, i.e., $\text{sign}(dx/dt)$ for input variable x . Circuit schematic in Fig. 9(b) computes sign of derivative of its input. Based on clock signals CLK/CLKB successive samples of x are sensed and compared to determine sign of its derivative. Fig. 10 shows the architecture of stochastic neural network for optimization. The network follows computation steps in [5], however, with stochastically controlled cost and constraints variables. Fig. 10(b) demonstrates transient evaluation of optimal solutions for primary and dual problem. In conjunction with stochastic decomposition, architecture in Fig. 10(a) computes the optimal solution minimizing expected cost for an optimization problem under uncertainty.

Conclusions

We have discussed a CMOS-based mixed-signal architecture for solving optimization problems under uncertainties. OTA-based Kernel implementation is

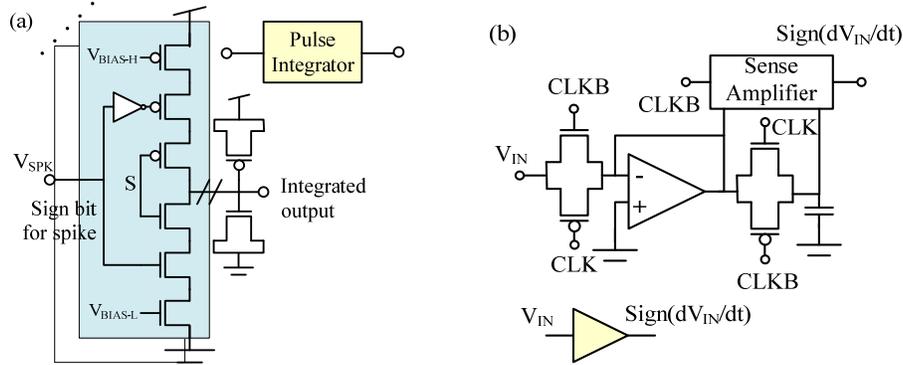


Figure 9: (a) Pulse integrator circuit. (b) Circuit module to determine sign of derivative of input variable, i.e., sign of dV_{IN}/dt .

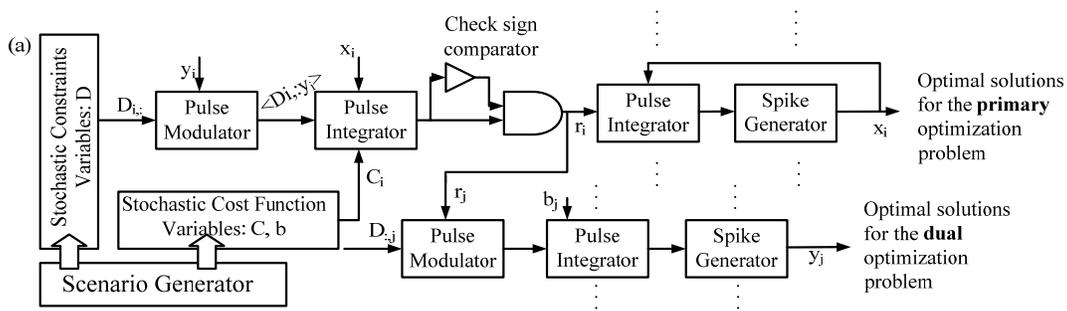


Figure 10: Architecture of stochastically spiking neural network.

discussed for on-chip learning and mimicking of statistics of uncertain (random) variables in the optimization problem. The discussed mixed-signal stochastically spiking neural network minimizes area of computing cells while facilitates easier integration and scalability.

Acknowledgement:

A. Trivedi would like to acknowledge startup research funds from University of Illinois at Chicago. S. Mukhopadhyay would like to acknowledge National Science Foundation CAREER Award and Office of Naval Research Young Investigator Program for financial support.

References

1. Kali et al. *Stochastic programming*. New York: Springer, 1994.
2. Kleywegt et al. *SIAM Journal on Optimization*, 2002.
3. Higle et al. *Mathematics of operations research* 16.3 (1991): 650-669.
4. Gratz et al. "Operational transconductance amplifiers." *Synth stromeko* (2005).
5. Hasan et al., "An efficient simplified neural network for solving linear and quadratic programming problems." *Applied Mathematics and Computation* 175.1 (2006): 452-464.