The Impact of Analog Computational Error on an Analog Boolean Satisfiability Solver

David A. Basford, Jared M. Smith, R. Joseph Connor, Bruce J. MacLennan, and Jeremy Holleman
Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, Tennessee 37996
Email: {dbasford,jms,rconnor6}@vols.utk.edu, {maclennan,jhollema}@utk.edu

***Abstract*—We present an analog implementation of a dynamical system for solving Boolean satisfiability, an NP-complete problem. Simulations of modest-sized hardware implementations in the presence of noise and integrator offset demonstrate that the algorithm is suitable for implementation in analog electronics.**

I. Introduction

A. Boolean Satisfiability

The Boolean Satisfiability problem (SAT) is to find a set of 0/1 assignments to the Boolean variables $X_1, \ldots, X_n$ so that a specified Boolean expression is true (= 1). The Boolean expressions are in *conjunctive normal form*, e.g.:

$$(X_1 + X_3 + X_4) \cdot (\overline{X_2} + X_3 + X_4) \cdot (X_2 + X_4 + \overline{X_5}).$$

In this instance, there are $N = 5$ variables and $M = 3$ *clauses*. Positive or complemented variables (e.g., $X_2, \overline{X_2}$) are called *literals*. If every clause contains exactly k literals (as in the above example, $k = 3$), then it is an instance of the k-SAT problem. k-SAT is an NP-complete problem.

The difficulty of a particular instance is related to the *constraint density*, $\alpha = M/N$ [1].

B. ACTNN-k-SAT Algorithm

Ercsey-Ravasz and her colleagues present a continuous-time dynamical system for solving k-SAT [2], [3]. There are N bounded variables s_i which evolve to a solution, positive for Boolean 1, and negative for Boolean 0. In addition, there are M bounded variables a_m that measure the "urgency" of satisfying a clause. A particular problem instance is represented by an $M \times N$ constraint matrix $\mathbf{c}$, where $c_{mi} \in \{-1, 0, 1\}$. $c_{mi} = +1$ if X_i is positive in clause m, $c_{mi} = -1$ if $\overline{X_i}$ is in clause m, and $c_{mi} = 0$ if X_i does not occur in clause m.

The dynamical system is defined by the differential equations:

$$\dot{s}_i(t) = -s_i(t) + Af[s_i(t)] + \sum_{m=1}^{M} c_{mi} g[a_m(t)],$$

$$\dot{a}_m(t) = -a_m(t) + Bg[a_m(t)] - \sum_{i=1}^{N} c_{mi} f[s_i(t)] + 1 - k.$$

The *self-coupling parameters* A and B are two real constants that depend weakly on k. The f and g activation functions are piece-wise linear squashing functions that map the s and a values into $[-1, 1]$ and $[0, 1]$, respectively:

$$f(s) = \begin{cases} -1 & \text{if } s < -1, \\ \text{s} & \text{if } -1 \le s \le 1, \\ +1 & \text{if } s > 1. \end{cases}$$

$$g(a) = \begin{cases} 0 & \text{if } a < 0, \\ \text{a} & \text{if } 0 \le a \le 1, \\ +1 & \text{if } a > 1. \end{cases}$$

Molnár and Ercsey-Ravasz [2] prove that the only stable fixed points of the system are solutions to the problem, and they give numerical evidence that there are no limit cycles, provided that the A and B parameters are in the appropriate range. Typical good ranges for the constants are $1 < A < 2$ and $2 < B < 2\lfloor k/2 \rfloor + 2$. If in fact there are no limit cycles in these ranges, then unsolvable instances would have chaotic attractors. This is consistent with the observed transient chaotic behavior for hard instances [2].

Molnár and Ercsey-Ravasz [2] prove the following bounds on the variables:

$$|s_i(t)| \le 1 + A + \sum_m |c_{mi}|,$$

$$-2k \le a_m(t) \le 2 + B,$$

provided that they are initially in the ranges $|s_i(0)| \le 1$ and $0 \le a_m(0) \le 1$. This guaranteed bounding of variables is critical for analog implementations, because variables in analog computational systems are bounded by their physical representation. The initial values are otherwise arbitrary.

The progess toward solution can be tracked by the following "energy function" which depends on the number of unsatisfied clauses [1], [2]:

$$E[f(\mathbf{s})] = \mathbf{K}^{\mathrm{T}}\mathbf{K} \text{ where } K_m = 2^{-k} \prod_{i=1}^{N} [1 - c_{mi} f(s_i)].$$

Note that E is not a Lyapunov function, and energy does not decrease monotonically.

Analog Implementation of ACTNN-k-SAT.

C. Analog Implementation of ACTNN-k-SAT

Fig. **??** displays an analog algorithm for implementing this dynamical system. The overall structure is a cross-bar between

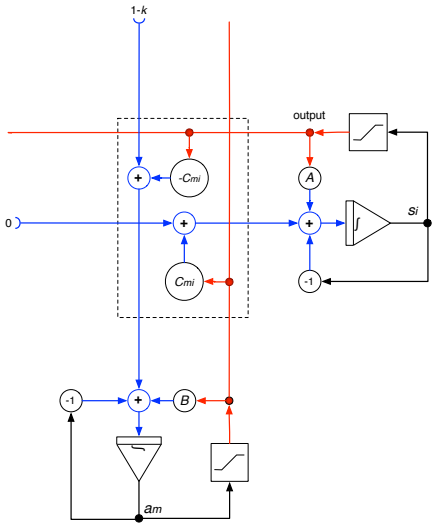

Fig. 1. ACTNN-k-SAT

the M integrators for the a_m and the N integrators for the s_i; thus $M + N$ integrators are required. A particular instance is programmed by setting the c_{mi} and $-c_{mi}$ connections to $-1, 0$, or $+1$, as required for the problem. The integrators are initialized to small values to start the computation; non-zero offset or noise in the hardware integrators might have the same effect.

Analog computation has been demonstrated to provide outstanding energy efficiency at moderate resolutions, often yielding energy efficiency improvements of multiple orders of magnitude over digital implementations [4][5]. However, with that efficiency comes imperfect computation, as every element injects error into a calculation. These errors can be reduced, but at the cost of increased silicon area and power consumption. In order to ensure that circuit performance does not degrade algorithmic performance, while avoiding excessive power and area due to over-engineering, designers must carefully model the effects of non-ideal computation. In this paper, we address that modeling task for the ACTNN-k-SAT solver, focusing primarily on noise and offsets.

D. Transistor Implementation

The design was implemented for simulation as a four-by-four array in an standard 130 nm CMOS process. The circuit is capable of solving problems with up to a maximum of four each of variables, clauses, and variables per clause. Larger problems may be solved by increasing the dimensions of the array. The circuit is fully differential, and the majority of the junctions were implemented as current summing nodes to minimize voltage swings. Current mirrors were used extensively to copy the current outputs of each cell to the inputs of the junctions across the array. The circuit operates at a V_{DD} of 1 V and consumes an average of 141 µW. The unit of differential current was chosen to be 100 nA.

The integrator was formed using a fully differential OTA with common mode feedback. A pair of 100 fF feedback

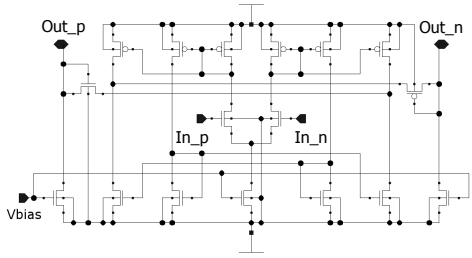

Fig. 2. Schematic of $g(a)$ cell

capacitors was used to achieve a slope of 7.69 mV/µs for a 1 nA differential input. A pair of series diodes from each input to ground provided the common mode bias voltage for the OTAs inputs.

The $f(s)$ function was implemented using an NFET differential pair. The bias current was provided by an NFET current sink which limited the output current swing when the inputs were driven beyond the linear range.

The $g(a)$ functions implementation was similar to that of $f(s)$ except that differential output current was only allowed to change for positive input. The schematic is shown in Fig. 2. This was accomplished by splitting the differential and common mode output currents from the differential pair. The differential current was then recombined with the common mode current through a pair of MOSFET diodes. At full positive input an NFET diode allowed up to 50 nA to be added to the positive current output, and a PFET diode would subtract up to 50 nA from the negative output. For negative input the diodes were cut off, so only the common mode current remained at the output.

The outputs of $f(s)$ and $g(a)$ are scaled by the constants A and B respectively to be used as feedback around their respective integrators. This was performed using scaled current mirrors. The values of A and B are 1.3 and 2.3 respectively as these were determined to be within the optimal range for solution accuracy and time [2].

The negative unity feedback around the integrators required conversion from voltage to current, so it was implemented using a linearized transconductor. The circuit is similar to that of the f cell except that the current sinks have been split and a resistor attached between the sources of the differential pair. The common mode current was increased to allow a larger output current swing, and the transistor sizes were adjusted so that the slope was close to the linear region of the f cell. The output still saturates for a large enough input, but the limit is much larger compared to the f cell.

The c_{mi} function was implemented using crossbar switches to connect a pair of NFET current mirrors alternately to the outputs. The inputs to the current mirrors are from the appropriate f or g signals. Two logic bits were used to control the switching action. The enable bit c_{mi_en} determines if the variable exists in the clause, and the polarity bit c_{mi_pol} determines whether or not the variable is complemented. If c_{mi_en} and c_{mi_pol} bits are both high, then the differential

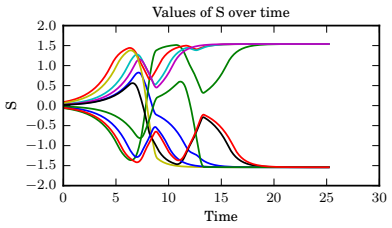

Fig. 3. Evolution of s variables over time for a typical $\alpha = 4$ 4-SAT problem.

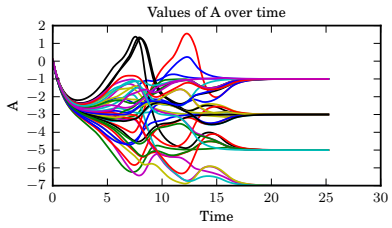

Fig. 4. Evolution of a variables over time for a typical $\alpha = 4$ 4-SAT problem.

current is allowed through unaltered. If the c_{mi_en} is high and c_{mi_pol} is low, then the current outputs are swapped. If c_{mi_en} is low, then c_{mi_pol} is ignored and the outputs of both current mirrors are split between both outputs. This effectively cancels the differential current output and leaves the common mode unchanged to avoid upsetting bias conditions of the following stage.

The $1-k$ input was formed by using a linearized transconductor like before except that a fixed 100 nA differential current is subtracted from the output. The outputs were then swapped to achieve the $1-k$ function. The slope for a differential input is 100 nA$/70$ mV. The common mode current of 435 nA allows values of k up to 5.

The 0 input shown on the left side of Fig. **??** can actually be used as a means of energy input to the system. Another f cell was used to provide input to the row blocks. A pulse at this input will cause the row integrators to leave their initial metastable states.

II. Simulation

Previous simulations have demonstrated that the algorithm is resilient in the face of significant noise in the interconnection weight c_{mi} and integrations [3]. In preparation for an eventual implementation in analog electronics, we developed a simple simulation program in order to evaluate performance of the algorithm. For the small instances of k-SAT that we have in mind ($N \approx 10, M < 50$) we use a simple forward-Euler method. We established that a time step $\Delta t = 0.02$ gives reliable results, and it was used in the simulations reported here. We focused on 4-SAT problems and used self-coupling parameters $A = 1.54$ and $B = 2.18$, which are in the optimal region established by Molnár and Ercsey-Ravasz [2]. Simulations were run for a maximum of $t_{\max} = 100$ time units, and were deemed to have failed to find a solution if unable to do so within this time.

Fig. 3 displays the evolution of the ten s variables over time for an example 4-SAT problem with a constraint density $\alpha = 4$. The graph show how some variables tentatively assume values early in the simulation, but change them later as the system is attracted to a solution state. Fig. 4 displays the evolution of the forty a variables for the same simulation. Careful comparison with Fig. 3 shows that increase in an a variable (corresponding to an unsatisfied constraint) leads to a sign change in an s variable. Finally, Fig. 5 shows the time

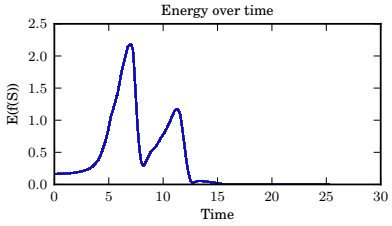

Fig. 5. Evolution of E "energy" function over time for a typical $\alpha = 4$ 4-SAT problem.

evolution of the energy function, which reflects the urgency of satisfying unsatisfied constraints. A partial solution is found at $t \approx 8$ before a complete solution is discovered at $t \approx 18$.

Simulations of the transistor circuit led to some interesting observations. Since all components were ideal and perfectly matched, the possibility of a metastable state exists. This means that if the outputs of the integrators are initially at zero, then they would remain there unless energy was input into the system. The predicted metastable state was not perfect though. The s outputs eventually swing negative due to slight mismatch in the slopes of the f, g, and transconductor cells. The perfect matching also meant that some combinations of c_{mi} inputs would result in oscillations at the output. One such example is shown below.

$$(X_1 + X_2) \cdot (\overline{X_1} + \overline{X_2}) \cdot (X_3 + X_4) \cdot (\overline{X_3} + \overline{X_4})$$

This resulted in all outputs simultaneously oscillating in phase. The solution requires that X_1 and X_3 be opposite respectively to X_2 and X_4. However, if the outputs are oscillating in phase, then they will not reach a stable solution unless some noise or offset is present in the circuit.

III. Results

Fig. 6 displays the distribution of convergence times versus constraint density α for 100 randomly chosen constraint matrices for 4-SAT instances with no noise or offset. For $\alpha < 3.5$, all instances are solved in the allotted time, but for $\alpha \geq 3.5$ a few remained unsolved when the simulation was terminated. Therefore, in the following simulations we used $\alpha = 4.0$ to provide a baseline against which to compare the effects of noise and offset.

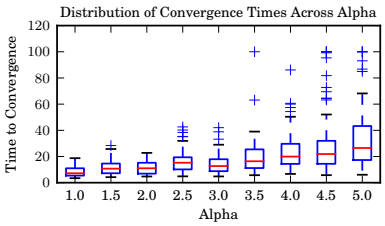

Fig. 6. Distribution of convergence time versus difficulty measured by α for 100 random 4-SAT instances. ($t = 100$ represents non-convergence in the allotted time.)

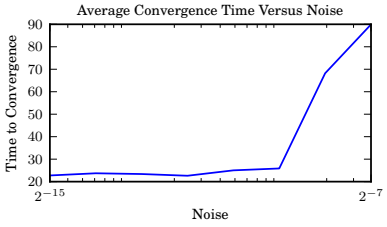

Fig. 7. Convergence time versus noise for 100 random 4-SAT instances.

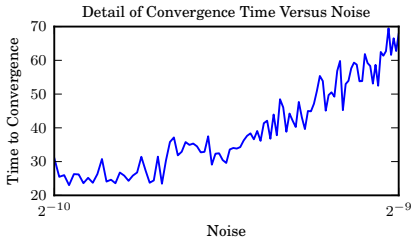

Fig. 8. Detail of convergence time versus noise for 100 random 4-SAT instances, noise $\sigma = 2^{-10}$ to 2^{-9}.

Fig. 7 displays the time to convergence for the same 100 random 4-SAT problems, but with Gaussian noise σ varying from 2^{-15} to 2^{-7} of full range of the s and a variables. Noise had little effect on the system up through values of 2^{-10} (nearly 100% solved). At 2^{-9}, however, only about 40% of the problems were solved in the time set for the simulation. Additional noise levels were simulated in range of $2^{-10} - 2^{-9}$ where the critical transition seems to occur, as illustrated in Fig. 8

The effects of noise and offset on convergence are summarized in Fig. 9, which displays the percentage of this random selection of $\alpha = 4$ problems that were solved within the allotted time for a variety of combinations of noise and offset.

The transistor circuit exhibited excellent speed. A typical 2-SAT problem was solved in less than 25 µs. Inserting a 100 mV, 10 ns pulse at the energy input terminal reduced the solution time to 15 µs. It was noted that the solution time was inversely proportional to the logarithm of the energy input, so using a large energy pulse is inefficient for reducing solution time. Although the quiescent power consumption is

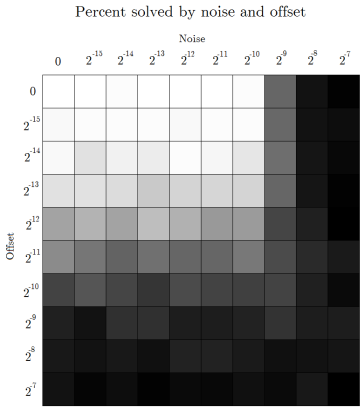

Fig. 9. Percent of $\alpha = 4$ problems solved for various combinations of noise and offset. Key: white = 100%, black = 0%.

high at 141 µW, the dynamic energy required to solve a 2-SAT problem is only 8.1 pJ. It is clear from this distinction that there is still much room for improvement in this circuit in regards to total power consumption. Rigorous investigation of the tradeoffs of speed and solution accuracy versus power consumption could lead to better optimization.

IV. Conclusion

We have determined that this analog algorithm for Boolean satisfiability can solve modest-sized instances ($N = 10, \alpha = 4$) in the presence of moderate noise and offset error, and is thus suitable for implementation in analog electronics. Further study is required to determine whether the noise requirement can be relaxed further with longer running time. More relaxed noise budgets could allow faster and more efficient implementations, reducing the scaling factor relating the arbitrary time units used in these simulations to real-world time. Therefore even though solution times might increase in simulation with noisier circuits, the real-world solution time could decrease. A four-by-four transistor version of the circuit was presented here, though a larger array could be built to solve larger problems.

References

[1] M. Ercsey-Ravasz and Z. Toroczkai, "Optimization hardness as transient chaos in an analog approach to constraint satisfaction," *Nature Physics*, vol. 7, pp. 966–970, 2011.

[2] B. Molnár and M. Ercsey-Ravasz, "Asymmetric continuous-time neural networks without local traps for solving constraint satisfaction problems," *PLoS ONE*, vol. 8, no. 9, p. e73400, 2013.

[3] R. Sumi, B. Molnár, and M. Ercsey-Ravasz, "Robust optimization with transiently chaotic dynamical systems," *EPL (Europhysics Letters)*, vol. 106, p. 40002, 2014.

[4] J. Lu and J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 µm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, February 2014.

[5] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz, 128×32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*. IEEE, 2004, pp. 651–654.