Importance Sampled Circuit Learning Ensembles for Robust Analog IC Design

Peng Gao[1], Trent McConaghy[1,2], Georges Gielen[1]
1 ESAT-MICAS, K. U. Leuven, Kasteelpark Arenberg 10, Leuven, Belgium
2 Solido Design Automation Inc., Saskatoon, Canada
{Peng.Gao, Trent.McConaghy, Georges.Gielen}@esat.kuleuven.be

ABSTRACT

This paper presents ISCLEs, a novel and robust analog design method that promises to scale with Moore's Law, by doing boosting-style importance sampling on digital-sized circuits to achieve the target analog behavior. ISCLEs consists of: (1) a boosting algorithm developed specifically for circuit assembly; (2) an ISCLEs-specific library of possible digital-sized circuit blocks; and (3) a recently-developed multi-topology sizing technique to automatically determine each block's topology and device sizes. ISCLEs is demonstrated on design of a sinusoidal function generator and a flash A/D converter, showing promise to robustly scale with shrinking process geometries.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles, B.7.2: Design Aids

1. INTRODUCTION

As Moore's Law [1] captures, the minimum size of transistors in integrated circuits has been decreasing exponentially for several decades. For digital design, the incentive to shrink geometries is high: smaller area, higher speed, and lower power. However, scaling is less beneficial to analog circuits because mismatch, which limits performance of many analog circuits, worsens as geometries shrink [2]. To cope, designers increase device area [3], use many circuit-level techniques like feedback and differential design [4][5], shift functionality to digital, and apply calibration. But these only partially scale with Moore's Law because large analog-sized transistors must form the core signal path. As a result, the analog portion of mixed-signal chips risks dominating area [6] and may even hit a wall when tuning is no longer enough [7]. We ask: is it possible to design analog circuits that are simultaneously (i) *naturally* robust to variation without needing tuning, yet (ii) *scale with Moore's Law*, i.e. use the smallest-possible transistors?

The answer we explore leverages two advances: (i) because the tiniest transistors have become so small relative to a typical analog transistor, we can potentially use far more transistors than a typical analog circuit, i.e. we can *waste transistors* [8], and (ii) recent developments in machine learning point to a new paradigm for designing analog machines – learning ensembles. The approach that we propose, ISCLEs, combines a lavish number of tiny transistors into a boosted ensemble of circuits.

The paper is organized as follows. Section 2 reviews machine learning, then describes the ISCLEs boosting algorithm developed for circuits. Section 3 describes the ISCLEs-specific library of digital-sized circuit blocks, and section 4 briefly describes the approach to determine each subcircuit in the ensemble. Section 5 describes experiments in designing a sinusoidal function generator and an A/D converter. Section 6 concludes.

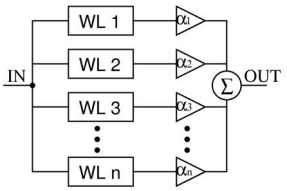

Figure 1: An ISCLEs circuit combines many weak learner circuits (WLs) to achieve a target analog functionality

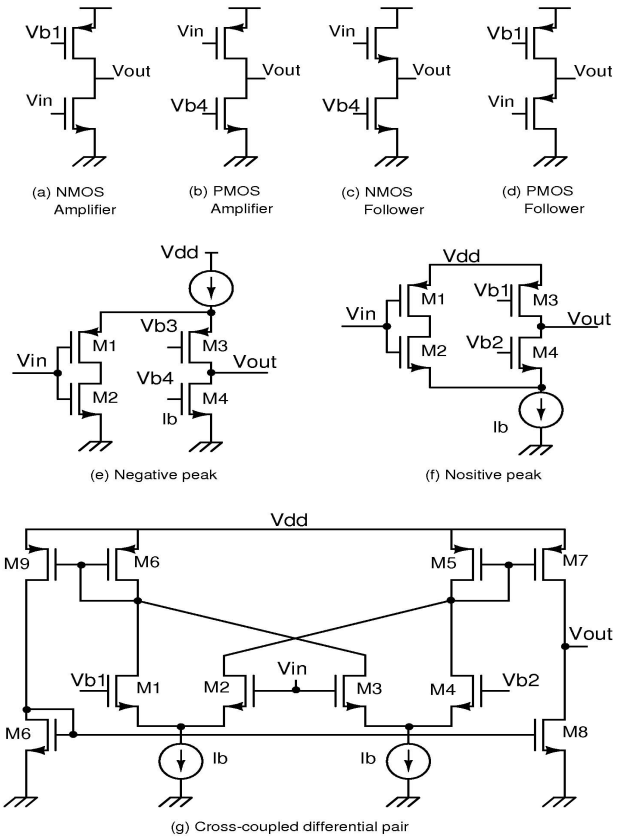

Figure 2: Weak Learners Library

2. MACHINE LEARNING AND ISCLES

In machine learning [9], the aim of supervised learning is to find an input-output mapping that predicts well on unseen data. For decades, the approach was to determine a single good model. This usually had the issue of overfitting, in which the model performed well on training data but generalized poorly to unseen data. But recently, *ensembles* of models [10], which combine the output of many learners, have been shown to overfit less because

errors made by sub-learners can be averaged out. In "bagging", each sub-learner learns the full input-output mapping. Alternatively, a series of "weak learners" can be "boosted" into an overall "strong learner" [11]. Weak learning is much easier to do than strong learning of one model: each learner only needs to do better than random, rather than fully capture the mapping [12]. An outer boosting algorithm combines the weak learners. Boosting does importance sampling in model space, hence the label Importance Sampled Learning Ensembles (ISLEs) [13].

From a machine learning perspective, existing analog IC design approaches focus on some single "strong" circuit to realize the target functionality. In contrast, ISCLEs is an ensemble of "weak" circuits, boosted to collectively realize the target functionality. Crucially, these weak circuits each have small area (via near minimally-sized transistors) so that overall area is not prohibitive. The overall architecture is shown in Figure 1, and each weak learner is one of the possible topologies in Figure 2.

Table 1: ISCLEs Algorithm

Input: $\boldsymbol{y_{overall,target}}$, r_{target}, α
Output: EL_{chosen}
$\boldsymbol{y_{current,target}} = \boldsymbol{y_{overall,target}}$
$r_{current} = 0.0$
$EL_{chosen} = Ø$
while $r_{current} < r_{target}$: #for each boosting iteration
  WL_{cand} = find-weak-learner($\boldsymbol{y_{current,target}}$)
  $EL_{cand} = EL_{chosen} + α \bullet WL_{cand}$ #(as a circuit ensemble)
  $\boldsymbol{y_{cand}}$ = simulate(EL_{cand})
  r_{cand} = correlation($\boldsymbol{y_{overall,target}}$, $\boldsymbol{y_{cand}}$)
  if ($r_{cand} > r_{current}$): #improved
    $r_{current} = r_{cand}$
    $EL_{chosen} = EL_{cand}$
    $\boldsymbol{y_{current,target}} = \boldsymbol{y_{overall,target}} - \boldsymbol{y_{cand}}$

Table 1 describes the high-level algorithm. The key input is an overall target waveform $\boldsymbol{y_{overall,target}}$, and the output is an ensemble EL_{chosen} to realize the waveform. At each ISCLEs boosting iteration, a weak learner WL_{cand} topology & sizing is chosen, and if it improves overall correlation, $r_{current}$, then it is added to the final ensemble with a weighting factor α, and the target waveform $\boldsymbol{y_{current,target}}$ gets updated. Over time, the target waveform shrinks, zooming in on the hardest-to-capture parts of the mapping. The loop repeats until stopping criteria is hit, at which point the ensemble is returned. The whole process is automatic.

Boosting learning rate, α, is set to 0.10, meaning that on each iteration, 10% of the newest weak learner's output is used to update the overall target waveform. This setting strikes a compromise between risk of overfitting (higher α), and slower convergence (lower α). Target correlation r_{target} is set to 0.95.

3. WEAK LEARNERS

3.1 Weak Learners Library

A central challenge in this work was to design a competent library of possible weak learners. Some applications may only need a simple inverter, and others may need more complex topologies. We designed three weak learners: an inverter, an inverter with I-V amplifier, and an inverter cross-coupled differential pair. They form the library of possible topologies that are traversed in multi-topology optimization. We now describe each weak learner.

3.1.1 Inverter Learner

This is the simplest weak learner. A top-level inverter instantiates as one of the sub-blocks shown in Figure 2(a) to (d).

3.1.2 Inverter with I-V Amplifier

Instantiations of this weak learner are in Figure 2(e) and (f). Its core idea uses the fact that current flow in an inverter is not a monotonic function of the input voltage. While the input sweeps from 0 to Vdd, the current will increase because the NMOS is gradually turned on, but after a certain threshold point, the PMOS switches off and current will reduce to 0 again. This will form a current peak whose position and width are determined by the two transistors' sizes. If the NMOS' aspect ratio is increased, the peak position will be lower, and vice versa. We then use an I-V amplifier to convert this current peak into a voltage peak; and via sizing we can control voltage peak waveforms. The peak's minimum width is limited by the finite gain and sensitivity of the I-V amplifier. A peak simulation result is shown in Figure 3, which shows how different waveforms between transition points are realizable by different sizings.

3.1.3 Cross-Coupled Differential Amplifier

This weak learner circuit, shown in Figure 2(g), is composed of a cross coupled differential pair and several current mirrors. The input signal is connected to one of the input pins of each differential pair. The other input pins are connected to different bias voltages V_{b1} and V_{b2}, which set two fixed threshold points [5]. The size of input transistor pairs controls the threshold points, such that the output transfer curve will be like Figure 3.

3.2 Weak Learner Parameters

Table 2 enumerates the parameters for all weak learners. Note that the maximum device size is just 20 times the minimum feature size, which forces the building blocks to be as small as digital circuits, i.e. to *enable analog circuits scaling*.

Table 2: Parameters for Topology Choice & Sizing

Weak Learner	Parameter Names	Parameter Range
Inverter Learner	W_{in}, W_{load}, L_{in}, L_{load}	[W_{min}, 20*W_{min}] [L_{min}, 20*L_{min}]
Inverter with I-V Amplifier	W_1, W_2, W_3, W_4, W_5, L_1, L_2, L_3, L_4, L_5	[W_{min}, 20*W_{min}] [L_{min}, 20*L_{min}]
Cross-Coupled Differential Pair	W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8	[W_{min}, 20*W_{min}] [L_{min}, 20*L_{min}]
Topology Choice	*Choice_index*	1, 2, 3, 4, 5, 6, 7

4. MULTI-TOPOLOGY SIZING

Each weak learner is found with MOJITO [14] searching the possible topologies and sizings. MOJITO views the search space as a parameterized grammar, then finds the optimal "sentences" with grammatical genetic programming [15]. MOJITO's objective is to maximize the correlation between the current target waveform(s) (as specified by the boosting loop) and its candidate

circuit's waveform(s). By optimizing on correlation rather than squared error, MOJITO's problem is easier because correlation ignores the difference in offset between waveforms; the outer boosting loop takes care of this with an offset voltage. MOJITO's constraints are device sizing constraints and device operating constraints (e.g. ``keep transistor in saturation'').

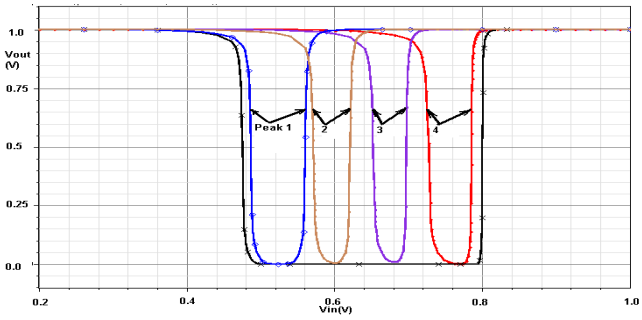

Figure 3: Four different negative voltage peaks (and bounds) of inverter with I-V-amplifier, via different sizings

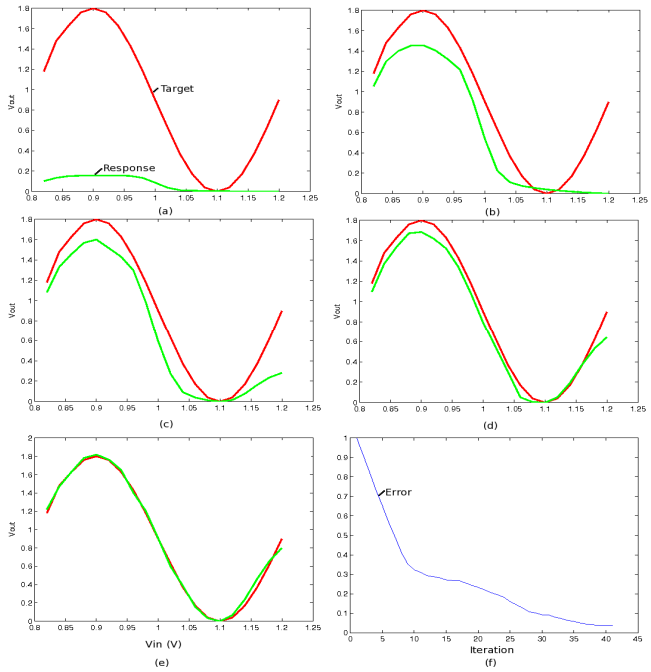

Figure 4: ISCLEs convergence on DC-DC sinusoidal circuit. (a, b, c, d, e) is target and response waveforms' V_{out} vs. V_{in} for iterations 1, 10, 20, 30, 40 respectively; (e) is error vs. iteration

MOJITO was configured to maximize search efficiency yet avoid getting stuck, using the following setup. At a given weak learner target, the population size was set to 10, and 50 generations were run. If the resulting circuit reduced the ensemble's overall error, then that weak learner was considered complete, and added to the ensemble. But if overall error did not improve, then the population size was doubled and MOJITO was re-run. In practice, we found that no doubling occurred in early iterations, but a few rounds of doubling occurred in later iterations. All other MOJITO settings were the same as [14].

5. EXPERIMENTS

We applied ISCLEs to two different kinds of problems: a DC-DC sinusoidal function converter, and a 3-bit flash A/D converter. The circuit simulator used was HSPICE™, using a 0.18µm CMOS process technology. All runs were on a single Linux machine with a single-core 2.0 GHz Intel processor. Other settings were given in sections 2-4.

5.1 Sinusoidal Waveform

In this example, ISCLEs is applied to generate a DC-in DC-out sinusoidal function generator. Specifically, the aim is to minimize the squared error difference between target DC response and synthesized circuit's DC response, for several different input DC values. Runtime was 8 hours.

Figure 4 shows the result of 40 boosting iterations, resulting in an ensemble of 40 weak learners. Sub-figures (a) to (e) show ensemble's output response waveform as it converges to match the target waveform. Sub-figure (f) shows the evaluation parameter NMSE (normalized mean squared error) vs. boosting iteration. We see that after 40 iterations, only a few percent error is remains between the target and ensemble circuit's responses. This example affirms the core idea of ISCLEs -- using boosting-style importance sampling to construct circuits.

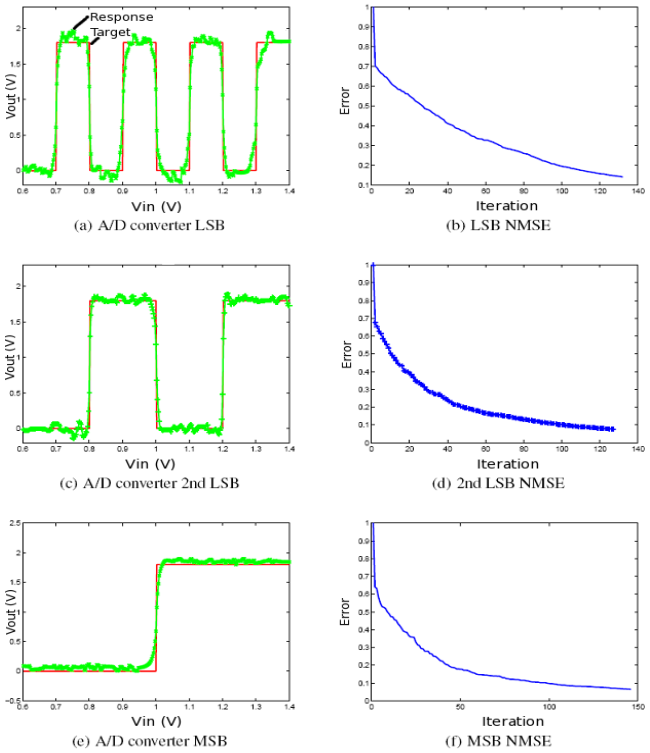

Figure 5: A/D design with ISCLEs. (a, b) is LSB, (c, d) is 2nd LSB, and (e, f) is MSB. For each bit, the left figure shows the output vs. input DC voltage, and the right figure shows the convergence of NMSE vs. boosting iteration.

5.2 3-bit Flash A/D Converter

The aim of this example is to target A/D conversion; we specifically aim for a 3-bit Flash architecture. Flash A/Ds are quite sensitive to process variations, due to the matching property of the resistor ladder and comparator [4]. We approach this problem by designing one bit at a time. For each bit, the aim is to minimize the squared error difference between target DC response and synthesized circuit's DC response, for several different input DC values. Runtime for all three bits was two days.

Figure 5 shows results. We observe that all the waveforms of the three output bits match with their target waveform within certain error margin. The LSB has the most complex input/output mapping, but ISCLES still achieved 13% error, having 131 weak learners. The 2nd LSB reached 9% error with 126 weak learners. The MSB also reached 9% error with 145 weak learners. Note that for actual implementation, the bits' outputs are usually passed through an inverter that would rail the outputs to the high or low voltage value (i.e. Vdd and ground), thus making the DC-DC mapping tighter yet.

5.3 A/D Converter Simulation with Process Variation

In this subsection, we test ISCLEs' ability to tolerate process variation by injecting variation into devices' V_{th}. Figure 6 shows four LSB Monte Carlo simulations with A_{VT} = 5mVμm. Notably, the response only changes slightly, indicating ISCLEs' potential to tolerate process variations.

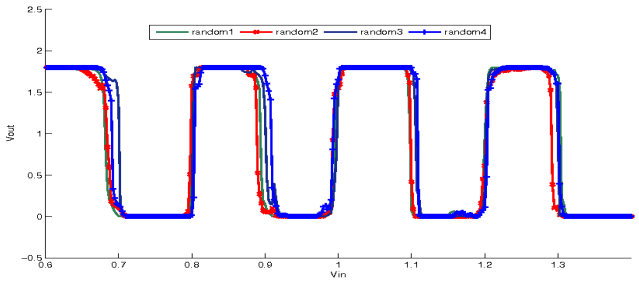

Figure 6: ADC's LSB with process variation injection

5.4 ISCLEs Scaling Potential

This section uses the A/D results to explore the potential of ISCLEs for scaling analog with Moore's Law. The ENOB of this 3-bit ADC @100kHz is 2.78. The estimated active chip area is 14e-09 m^2 (in 180nm CMOS) and 10e-09 m^2 (in 90nm CMOS). Then we predict the area of a conventional A/D [16], which should tolerate resistor matching (1%) and V_{th} variation (5mVμm), and achieves the similar ENOB. By rough estimation this chip should be larger than 1e-09 m^2 (in 180 nm CMOS) and 0.8e-09 m^2 (in 90 nm CMOS). According to the ITRS [2], A_{VT} will stop shrinking, but analog area will not shrink anymore [2]. (The chip will still slowly get smaller because of the shrinking of the digital part). With ISCLEs, mixed-signal chips will continue shrinking because the analog side uses minimally-sized transistors. By extrapolating as shown in Figure 7, we see that when technology shrinks to lower than 18nm, ISCLEs will become beneficial in chip area compared to conventional approaches.

6. CONCLUSION

This paper has presented a novel analog integrated circuit design method for robust design and with good technology scaling properties. It adapts boosting-style importance sampling from machine learning into the context of circuit design, by combining dozens of digitally-sized "weak learner" circuits to get an overall target analog functionality. To support the boosting framework, we designed a library of weak learner topologies; which are selected and sized at each iteration by multi-topology sizing. ISCLEs was demonstrated on two problems: a sinusoidal function generator, and 3-bit A/D converter learning. By demonstrating resilience to process variations yet using minimally-sized devices, ISCLEs has promise as a way for analog circuits to scale with process technology.

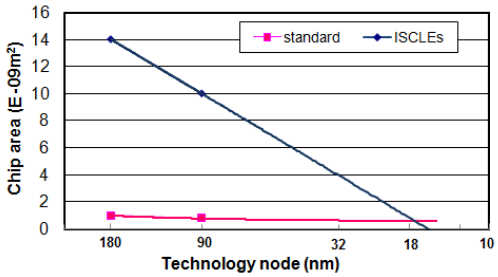

Figure 7: Prediction of chip area for 3-bit ADC using a standard design approach, versus using ISCLEs approach

7. REFERENCES

[1] G.E. Moore, Cramming more components onto integrated circuits. *Electronics Magazine 38(8), April 19, 1965*

[2] "International Technology Roadmap for Semiconductors," 2007. http://public.itrs.net

[3] M.J.M. Pelgrom, et al., "Matching Properties of MOS Transistor", *IEEE JSSC* 24 (5), Oct. 1989, pp. 1433-1440

[4] W.M.C. Sansen, Analog Design Essentials. Springer, 2006

[5] D. Johns & K. Martin, *Analog Integrated Circuit Design*, *Wiley* 1997

[6] R. Rutenbar et al., "Panel: Will Moore's Law Rule in the Land of Analog?", *In Proc. DAC 2004*, pp. 633-633

[7] G. Gielen et al, "Panel: Analog and digital circuit design in 65 nm cmos: End of the road?" *Proc. DATE*, 2005, *pp. 36-42*

[8] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980

[9] T. Hastie, R. Tibshirani, J. H. Friedman. The Elements of Statistical Learning. Springer, 2001

[10] R. Polikar, "Ensemble Based Systems in Decision Making," *IEEE Circuits and System Magazine*, Third quarter 2006

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting." *J. Comp. & System Sci.*, 55(1), 119–139, 1997.

[12] J. Friedman, "Stochastic Gradient Boosting," *J. Comp. Statistics & Data Analysis* 38(4), 2002, pp. 367-378

[13] J. Friedman, B. Popescu, "Importance Sampled Learning Ensembles", *Tech. Report, Stanford U. Dept. Statistics*, 2003

[14] T. McConaghy et al., "Simultaneous Multi-Topology Multi-Objective Sizing Across Thousands of Analog Circuit Topologies", *Proc. DAC*, 2007

[15] P.A. Whigham, "Grammatically-based Genetic Programming," *Proc. Workshop on GP: From Theory to Real-World Applications*, J.R. Rosca, ed., 1995

[16] R. v. d. Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd ed., Kluwer, 2003