Analog Integr Circ Sig Process
DOI 10.1007/s10470-012-9998-1

Operating-point driven formulation for analog computer-aided design

Ivick Guerra-Gómez · Trent McConaghy · Esteban Tlelo-Cuautle

Received: 24 February 2012 / Revised: 19 November 2012 / Accepted: 21 November 2012

Abstract In designing analog integrated circuits, the step of selecting device sizes and biases is crucial to enhance the final performance, power, and yield of the circuits. In manual design flows, the designer first selects bias voltage and branch current values, and then converts to widths and lengths via first-order equations. This is the operating-point driven (OP) formulation. Using OP makes it easy for the designer to maintain insight of the effect of design variables (voltage, currents) on performance, in addition to simplifying the sizing problem. In contrast to manual, automated sizing approaches in the computer-aided design (CAD) literature predominantly manipulate device sizes directly. While simpler from a CAD perspective, the direct-sizing approach misses the benefits of OP—insight and a simpler problem that would lead to faster convergence and better results. We believe that the OP formulation is underused, because of lack of familiarity, inconvenience, and underappreciation of its benefits. This investigation aims to resolve that and help to improve CAD practice, by reviewing the literature on OP that has been accumulated over the decades, and describing the variants, applications, and benefits of OP.

I. Guerra-Gómez
SEMTECH-Snowbush Mexico Design Center, Mexico, Mexico
e-mail: ivickguerra@yahoo.com.mx

T. McConaghy
Solido Design Automation, Saskatoon, Canada
e-mail: trentmc@solidodesign.com

E. Tlelo-Cuautle (✉)
INAOE, Luis Enrique Erro No. 1,
72840 Tonantzintla, PUE, Mexico
e-mail: etlelo@inaoep.mx

Keywords Operating-point driven formulation · Analog CAD · Biasing · Sizing

1 Introduction

Finding the sizes and biases of analog integrated circuits (ICs) remains a challenging problem, whether the approach to do so is manual or automated. In CMOS IC design, circuit sizing and biasing is a process through which the dimensions of the width (W), length (L), and multipliers (M) of the transistors are set, along with setting the supply voltages and reference currents. It is important because sizes and biases directly affect the performance, power, and yield of the circuit. It is challenging because the mapping from the sizes to the biases is often highly nonlinear, the design space is large (10's to 1,000's of design variables), and the feedback of the quality of the circuit can be time-consuming (e.g. via long SPICE simulations) or inconvenient (e.g. developing manual equations).

Sizing and biasing remain an ongoing challenge because there are always new process nodes with different characteristics (higher process variation, lower supply voltage, different MOS behavior), new target functionality, and new topologies. Due to this challenge, there has been a steady stream of efforts from the analog CAD community to help designers, by automating sizing and biasing. This can be observed in several design automation tools, for instance: IDAC[10], OP-1 [42], COARSE [19], OPASYN [26], ISAID [47], FRIDGE [35], DARWIN [27], ASTRX/OBLX [36], ANACONDA [37], WATSON [9], AMIGO [20] and OPERA [48] among others.

In the majority of cases, these CAD sizing approaches manipulate widths and lengths as independent design variables. This is especially true for the "simulation-in-the-

Published online: 11 December 2012

loop" approaches that have become popular since late 1990s. Each design candidate (or just the good candidates) is composed of widths (W's) and lengths (L's). Its performance is typically found by directly inserting the W's and L's into the netlist, then simulating with a SPICE-like simulator. Figure 1 depicts this "WL formulation."

On the other hand is the traditional manual sizing flow, which consists of several steps [38]. First, the designer defines the overdrive and bias voltage for each transistor. All the L's are set to the minimum value allowed for the technology and by using first-order models are found the W value for each transistor. The next step is setting a power budget and Vdd to compute the current budget. Then, all the currents are allocated among the different branches of the circuit. The next step is to keep adding constraint values for other performances until all the remaining I's and V's are set. With all the independent values resolved, first-order equations are used to compute each device width from the device's biases. The final step is to test the circuit in SPICE. Because the search space is I's and V's, the manual flow is a variant of the Operating-Point driven (OP) formulation, as shown in Fig. 2. If specifications are not met, then the designer either tweaks W's and L's, or goes back to the constraint-setting step and revises constraints.

While usage of OP is typically an "obvious" approach to those deeply trained in analog circuit design, it is not necessarily so to those in the analog CAD field. Analog CAD practitioners often have non-analog backgrounds, and tend to use the problems defined by other CAD practitioners. Given this, and that the WL formulation has a convenient direct mapping from design variables to evaluated netlist, the WL formulation may seen "obvious" to CAD practitioners.

However, it is entirely possible for CAD tools to exploit the OP formulation in automated sizing and biasing. The overall structure of the automated OP flow is shown in Fig. 2 the same overall flow as the manual OP flow. The idea is to use currents (as drain current of the transistor), voltages (as gate-source, drain-source or overdrive voltages of the transistors) and sometimes L's, for selecting the transistors' operating point. Each candidate design point is now voltages (V's) and currents (I's). To be evaluated, there is an intermediate step that converts the I's and V's into W's and L's for each transistor, using first-order equations, lookup tables, or local optimization on each

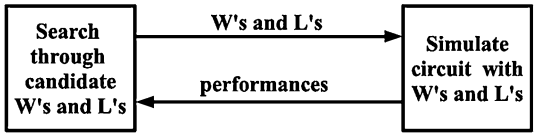


Fig. 1 WL formulation for sizing

Fig. 2 Operating-point driven (OP) formulation for sizing (variant using I's and V's)

device. Finally, the W's and L's are inserted into the netlist, and the circuit is simulated as usual.

Using an OP formulation in automated sizing can have several benefits: faster algorithm convergence, better-quality results, better designer insight, improved acceptance of automated sizers by designers, and more.

We believe that the OP formulation is underused, because of lack of familiarity, inconvenience, and underappreciation of its benefits. Also, the CAD researchers who have used it tend to underemphasize its importance. The contribution of this paper is to overcome these issues via a survey of the OP formulation. We will explain its benefits and its applications, while simultaneously reviewing a diverse set of papers and how they apply the OP formulation. It is our hope that use of the OP formulation becomes standard practice in analog CAD, to the benefit of the whole field.

The paper is organized as follows. Section 2 shows a circuit sizing by using WL and OP formulation. Section 3 explains the benefits of using an OP formulation. In Sect. 4 there are exposed several applications of the OP formulation. Section 5 describes the different OP formulation variants and a brief description of them. Section 6 is devoted to the related work to the OP formulation and the final section gives the conclusions.

2 Circuit sizing example

To illustrate an OP formulation lets us consider the simple current mirror depicted in Fig. 3. We have chosen a methodology that consists of characterizing the transistor behaviour from a gm/I_D versus $I_D/(W/L)$ plot [43]. If second order effects are neglected, then $gm/I_D = 2/V_{OV}$. The gm/I_D versus $I_D/(W/L)$ plot is used as a look up table to find the transistor size by using (1) [15]. It is performed by establishing an I_D budget and a V_{OV}.

$$\frac{W}{L} = \frac{I_D}{I_D/(W/L)} \tag{1}$$

Figure 4 shows gm/I_D versus $I_D/(W/L)$ for a 0.18 µu IC technology characterized with $L = 0.35$ µm and $W = 3.5$ µm.

To size the simple current mirror shown in Fig. 3, with $I_{ref} = 50$ µA, $L = 0.9$ µm, and to accomplish for both transistors a $V_{OV} = 0.2$ V $(gm/I_D = 2/V_{OV} = 10)$, then

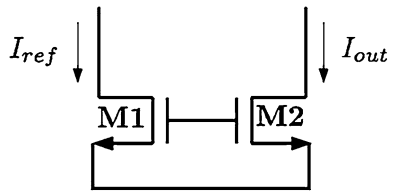


Fig. 3 Simple current mirror example

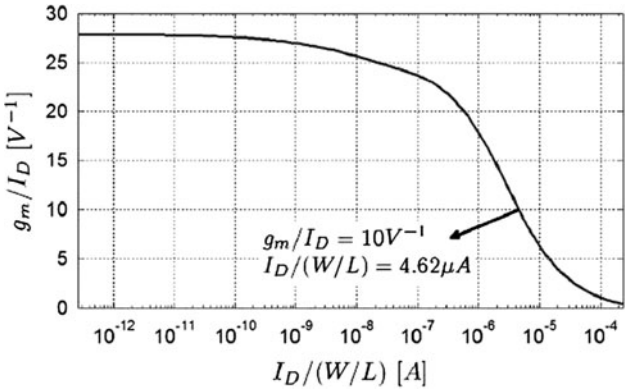


Fig. 4 gm/I_D versus $I_D/(W/L)$ for a 0.18 µu IC technology

from the gm/I_D versus $I_D/(W/L)$ depicted in Fig. 4, the corresponding value for $I_D/(W/L) = 4.62$ µA, leads to $W/L = 10.94$. By using (1), with $L = 0.9$ µm, one gets $W = 9.738$ µm (9.72 µm was used because it is a multiple of 0.18 µm). After performing the simulation with these size values, $V_{OV} = 0.194$ V, that is around 3 % in difference from the desired value of 0.2 V.

$$I_D = \frac{1}{2}\frac{W}{L}\mu_o C_{ox}(V_{gs} - Vth)^2 \quad (2)$$

On the other hand, performing the same experiment but by using the WL formulation, instead of the OP formulation, from (2) the value of $\mu_o\, C_{ox}$ is 342 µA / V^2, then $W/L = 7.31$. For $L = 0.9$ µm the width becomes $W = 6.57$ µm (6.66 µm was used because is a value multiple of 0.18 µm). After performing the simulation, $V_{OV} = 0.175$ V, that exhibits a difference about 12 % from the desired value of 0.2 V.

As one sees, the operating-point driven formulation is quite useful and its main benefits are gaining insight and dealing with a simpler problem than the WL formulation. Additionally, the OP formulation lead to faster convergence and better results. For this reason, this investigation aims to highlight the benefits of OP formulation to improve CAD practice. The rest of the paper reviews the literature on OP that has been accumulated over the decades, describing the variants, applications, and benefits.

3 Benefits of the OP formulation

We now review benefits of the OP formulation.

Faster convergence, better-quality results Perhaps most importantly, OP can make the sizing problem easier, resulting in faster algorithm convergence, and better-quality results. The main reason for this is because the I's and V's have fewer interactions when mapping to output values, compared to the complex interactions that W's and L's can have [28]. This makes a major difference in automated sizers, because if one can minimize the number of interactions among independent variables, the circuit sizing problem becomes much simpler. In some cases, having N voltage nodes and M transistors, the OP problem casts one problem of size N into a set of M 1-dimensional problems. In this scenario, the sizing algorithm can optimize for one design variable at a time, making for a near-trivial optimization problem.

OP may also make the problem easier because by creating fewer design variables, though the effect from that is smaller than the effect from fewer variable interactions. An intuitive example is: the DC currents in devices along the same branch get shared [12, 18, 23]. In [25] an OP formulation is applied by using "sizing and biasing operators," and the process takes a just few seconds to finish the sizing and biasing task. The authors of [7] report a 10^6 times reduction in the size of the search space, for an opamp with 16 design variables. This method was applied to synthesize a 14-bit ADC, reducing the power consumption more than 60 %. By using OP formulation, the authors of [34] found a 10 times reduction in the number of simulations needed to get similar results quality.

A new methodology has been introduced in [29]. It is based on interpolation and shows 12 times improvement on accuracy and more than 1,000 times improvement on efficiency compared with other techniques listed in that work.

Recently, an automated biasing and sizing technique was introduced in [24], where a fixed-point iteration avoids the derivative required in Newton Raphson based techniques. That technique is based on sizing and biasing operators originally proposed in [22].

Better designer insight, improved acceptance by designers Being more similar to a manual design flow, OP changes the same independent design variables (voltages and currents) as the designer is used to manipulating during design. Furthermore, the designer has first-order equations relating these variables to performance. Therefore, a designer using an OP-based automated sizing tool will find it easier to main insight into the design, and ultimately find it easier to adopt the tool. Also, simply providing the

operating point of a circuit front-and-center gives the designer tremendous insight [4]. For instance, manual sizing approaches, such as [46], are enabled via the drain current equation, which relates the current in the drain terminal, the voltages in the transistor terminal, and the width and length of the transistor. This can be viewed as an OP variant, fitting into Fig. 2.

More flexibility for designers Because OP approaches are more similar to the traditional manual sizing flow than the WL based sizing, designers may find them easier to exploit in conjunction with their other design techniques, including hierarchical design [7, 13] and topology selection / design [34].

Finding operating point more consistently Traditional WL approaches, especially those with a Newton-Raphson SPICE solver and global optimization, can have difficulty getting circuits in some regions of WL design space to DC-converge. Because its design space is naturally the DC values, the OP formulation does not have these issues [28, 29]. Also, the OP formulation can help to find near or optimal values to an operating point [3].

Directly handling device operating constraints An OP formulation can handle device operating constraints (DOCs) easily, by setting a feasible region for each one of the currents and voltages [45]. These constraints make the search space smaller, and avoid the need for expensive tests on whether they meet DOCs [34]. The authors of [11] reported a reduction of 99 % in the size of an 11-variable design space via DOCs. SyEnA [4] determines feasible voltage ranges where the operating point is preserved.

Variety of application While this paper focuses on the benefits of OP for automated sizing and biasing, there are in fact other applications which can also benefit from OP [12, 23], and even new opportunities for applications. The next section explores these applications.

4 Applications of OP formulation

Besides giving helping automated cell-level sizing, OP has many other applications. This section explores those applications.

Hierarchical automated sizing The work in [13] shows that using an OP formulation enables better-quality results and shorter runtime, in sizing hierarchically-organized circuits. It shows a hierarchical circuit design methodology, that leverages the proposed OP formulation [28] for sizing at the cell level. It tested the methodology on a third-order single-loop continuous-time delta-sigma modulator, optimizing power consumption, area, and SNR. CHAMS[25] also performed hierarchical optimization and exploited OP formulation.

Classification / Regression In [7], active sampling was used to adaptively search an input OP space, to automatically construct a one-class classifier that bounded possible analog circuit performances. The paper [11] used active sampling to construct a classifier mapping input currents and voltages to feasibility, followed by a regressor that mapped input currents and voltages to performance.

Symbolic modeling Symbolic Modeling generates analytical expressions that describe the performance of analog circuits as a function of design variables (or other input variables). Some works have used OP variables as inputs: [6] automatically constructs fixed-template symbolic models mapping input currents and voltages, to output performances; and [33] automatically constructs template-free symbolic models on the same data, improving on accuracy and compactness.

Topology selection / synthesis In [34], an algorithm automatically searched across 100,000 different circuit topologies, and associated currents and voltages (OP formulation) which got converted to sizes and biases on-the-fly using look-up tables.

Variation-aware automated sizing The ANACONDA [37] tool added safety margins to device operating constraints, in a WL sizing formulation. The authors showed that with safety margins, the optimized circuits were naturally more robust to process variations. It is entirely possible to use safety margins on the DOCs in an OP formulation (which amounts to slightly smaller design spaces), to achieve the same circuit-robustness benefits.

The OP formulation even enables new applications, like the following examples.

Portable models of circuit performance The authors of [1] proposed the construction of models that map from OP currents and voltages to performance, and went on to suggest that these models can be process-independent. Therefore the models are portable, and the only work when going to different processes is to build the mapping between WL and OP on a per-device basis.

New manual sizing flows In [3], a manually-oriented transistor level sizing is performed, based on the EKV transistor model and the inversion coefficient. The authors developed a graphical design interface, for the designer to interactively specify the circuit performances and constraints, and get graphical feedback. Reference [44] has a similar approach, allowing specification of EKV model parameters, supply voltages, and bias currents. It partitions the circuit into basic analog structures for sizing and

biasing: single transistors, current mirrors, differential pairs and OTA's.

Rapid technology porting The OP approach to migrate from an old to new process [16, 17, 39–41] explicitly maintains each transistor's operating point, to prevent performance degradation. The steps are: (1) measure the OP on old technology, (2) for each device on the new technology, find the W and L that meet the measured OP. Step 2 is performed on a per-device basis, using analytical first-order models, root-solving / optimization, or interpolation models / lookup tables. It is more accurate than simple scaling, and faster than local optimization.

5 Variants of OP formulation

5.1 W as a function of I's and V's

This approach [13, 28, 34] leverages the relation for a MOS device in (3)[1].

$$W_i = f(I_{DSi}, V_{GSi}, V_{DSi}, V_{BSi}, L_i) \tag{3}$$

Equation (3) emphasizes that if the device's voltages, drain current, and length are known, then its width can be computed. The f may be a simply Level-1 transistor model, which is invertible and therefore W is analytically solvable. More complex transistor models must be addressed with higher-order techniques such as root-solving / optimization [28], Hermite interpolation models [28], or lookup tables [34]. Root-solving / optimization is extremely fast, since it is simply M 1-d optimization problems; and interpolation or tables is even faster because no simulation is necessary (assuming the MOS models were pre-characterized).

In this formulation, the design variables are independent chord currents I_c, node voltages V_n, and device lengths L. At a given design point, branch voltages V_b are determined from node voltages $V_b = A^T V_n$, and branch currents I_b are determined from independent chord currents $I_b = B^T I_C$, where A is the incidence matrix and B the basic loopset matrix used in simulation [28]. These values become the inputs to Eq. (3), to determine W for each device. With W and L for each device, the circuit may be simulated, and specs for the design computed.

Another way to solve (3) is by using regression as in OIOPD [29]. In this case I_{DSi} is selected as the variable to do the interpolation with W_i. All the other variables ($\{V_{GSi}, V_{DSi}, V_{BSi}, L_i\}$) are fixed by directly performing on-line simulations. This method can enhance the accuracy, has no memory storage requirements (such as lookup tables) and has a less execution run time.

[1] Bulk effect is ignored for simplicity.

5.2 g_m/I_D-Based OP formulation

References [14, 15, 43] use the so-called g_m / I_D method, having transconductance (g_m) and drain current (I_D) at the core. It considers how the ratio g_m/I_D relates to the normalized drain current $I_\square = I_D/(L/W)$. The quantities g_m/I_D and I_D are directly related to circuit performances, transistor operating regions, and the transistor dimensions [43].

Figure 5 shows a typical g_m/I_D (manual) design flow. First, the designer allocates bias currents according to a power budget. Then, different values of g_m/I_D versus $I_\square$ are swept, and a particular value is chosen per transistor based on some specs that are a function of g_m/I_D (e.g. DC gain). From each choice, the W/L of each transistor emerges. Given the W/L's and other specs taken as constraints (e.g. area, stability, parasitic capacitances), the L's are calculated. Given the W's and L's, remaining specs are calculated. Automated versions of such a flow would treat bias currents, g_m/I_D, and $I_\square$ as independent design variables. This methodology exhibits some discrepancy for V_{OV} values that belong to strong inversion. The short channel effects such as the velocity saturation due to high lateral field and the mobility degradation due to high vertical field have to be added to the gm/I_D curve by taking into account those effects.

5.3 Inversion coefficient OP formulation

The paper [3] uses the concept of "MOSFET Operating Plane", which consists of a plane where the first axis is the inversion coefficient (IC), and the second axis is the transistor length. Figure 6 depicts this plane, and the tradeoffs among the circuit performances as a function of IC and L, for a transistor in saturation with a fixed I_D.

$$IC = \frac{I_D}{2nk(W/L)U_T^2} \tag{4}$$

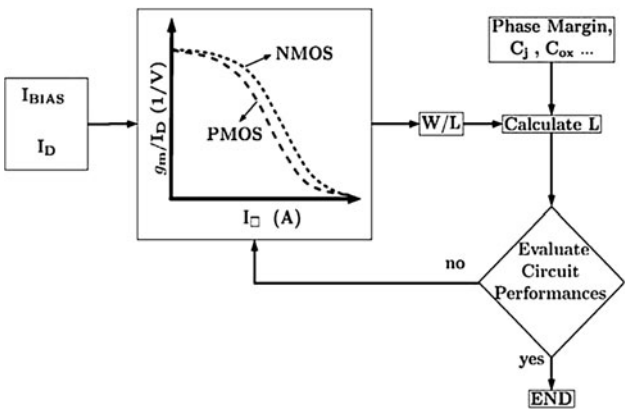


Fig. 5 g_m/I_D Methodology (from [43])

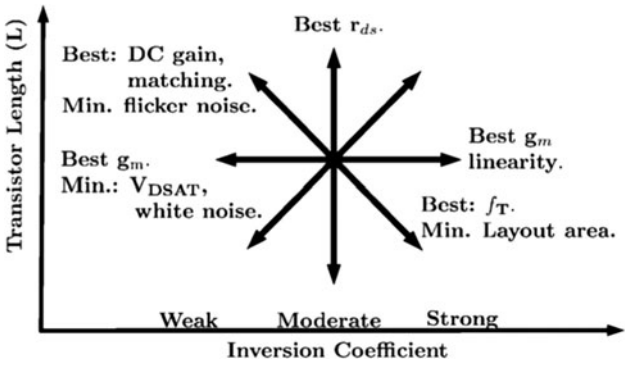


Fig. 6 MOSFET operating plane (from [3])

IC is calculated according to (4), where n is a parameter that relates the capacitive division between gate, surface and body, $k = \mu C_{OX}$ and $U_T = kT/q$ is the thermal voltage. The Operating Plane identifies the tradeoffs among the circuit performances and design parameters (voltages, currents and transconductances) with the *IC*. Table 1 shows some of those relationships, where I_0 is a technology parameter independent of the transistor bias condition and $IC_0 = LI_D/I_0W$. The W value is calculated in a follow-up step, taking into account the relationships among L, W, *IC* and I_D. The aim of using *IC* is to reduce the degrees of design freedom by using only L (instead L and W) for selecting the MOS operating point. It also helps the designer to select the operation around the weak, moderate or strong inversion, which directly impacts the circuit performances. While [3] used the MOS EKV model, the authors note that it easily extends to other MOS models.

CHAMS [25] performs a hierarchical sizing and biasing by using CAIRO+ [21]. This method uses the parameter mapping depicted in the blocks of Fig. 7. It uses as design parameters the V's, I's and L's (first block). Next, to solve the biasing and sizing problem (second block), it employs a Newton-Raphson algorithm based on (3) and an inversion equation like (4). The third block uses a transistor model with the aim to find the small signal parameters, and finally, the circuit performances are extracted from a circuit simulator (fourth block).

The procedure is based on biasing a single transistor. When all transistors are biased, the currents, voltages and small signal parameters are determined, mapping to linear

Table 1 IC and parameters relationships [3]

Parameter	$IC_0 \uparrow$ L, I_D fixed
$W = \left(\frac{L}{IC_0}\right)\left(\frac{I_D}{I_0}\right)$	$\downarrow \propto \frac{1}{IC_0}$
$V_{GS} - V_T = 2nU_T ln(e^{\sqrt{IC}} - 1)$	$\uparrow \propto ln(IC)$
$V_{DSAT} = 2U_T\lfloor(\sqrt{IC + 0.25} + 0.5) + 1\rfloor$	Unchanged

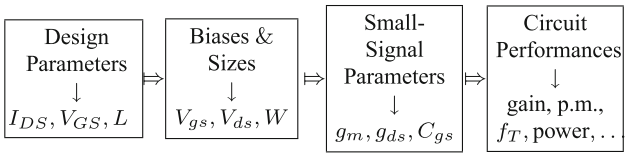


Fig. 7 Parameter mapping in the design space [21]

performances. The whole process is guided by a dependency-directed cyclic graph that is constructed in a hierarchical bottom-up fashion. In a final step, the operating point is verified with a simulator, and the design values are updated to achieve the desired specifications. In this manner, the methodology is capable of handling different MOS models, because the operation point is verified by the circuit simulator.

5.4 Graph-based OP formulation

Approach [7] works from a well-defined framework ("platform"), which consists of: a set of input variables, a behavioral model, a performance model (it is possible to have a performance model contained in the behavioral model) and validity laws. Circuit design is performed in a top-down fashion by using analog platforms that map higher level specifications to the lower level constraints [5]. There are topological and physical constraints. With each circuit, the topological constraints set the same widths, lengths or voltages to several transistors to match currents in the branches of the circuit. The physical constraints set the conditions or relations necessary to achieve the saturation condition in the transistors.

A behavioral model is used to approximate the performances of a given circuit. To achieve the sizing of the circuits, the model has topological and physical constraints added. Also, the authors relax the constraints by handling a tolerance range for each constraint to accomplish. As result they deal with a set of equations, a set of constraints and a set of variables by using Analog Constraint Graphs (AGC) [8] which allow reduction of the dimensions of the sampling space. An AGC is a representation of the configuration constraints and is a formal representation that generates random samples over the search space. Figure 8 depicts an AGC where circles represent design variables, rectangles equations and ovals constraints.

6 Related OP approaches

6.1 Relaxed DC formulation

Related to the OP formulation is the "Relaxed DC Formulation" [36] that approaches the sizing problem not by

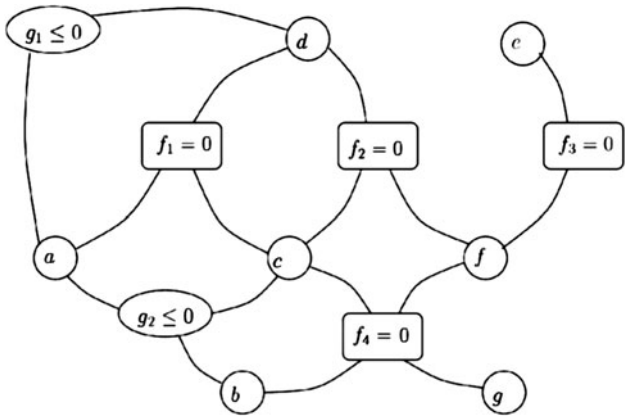


Fig. 8 Analog constraint graph (from [8])

simulation, but rather with constraints to meet Kirchhoff's current and voltage laws. Specifically, it renders the circuit design into a minimization problem of a weighted-sum cost function as shown in (5), where **x** is a vector representing the design variables, **f**(**x**) are the k circuit performances, **g**(**x**) are the l constraints and ω are scalar weights associated to each performance and constraint.

$$\min\left(\sum_{i=1}^{k}\omega_i f_i(\mathbf{x}) + \sum_{j=1}^{l}\omega_j g_i(\mathbf{x})\right) \tag{5}$$

The MINLP [32] approach is similar, but this time the minimization problem is based on minimizing a function subject to different constraints as performance goals , voltages, currents and design variable ranges. GPCAD [30] and its derivatives are similar, except with the additional constraint that the form of the equations is restricted to a convex optimization formulation.

6.2 Device operating constraints in WL formulation

Device Operating Constraints (DOCs) are constraints that specify the allowed voltages across terminals in a MOS or a circuit, and allowed currents in devices or circuit branches. DOCs can be used in both OP or WL formulations. As discussed previously, DOCs are naturally built into the OP formulation, because they are the bounds of the search space. But DOCs can also be used in the WL formulation, and are worth mentioning here because they still deal with the voltages and currents on devices.

A prototypical example of DOCs is the WiCkeD [2] tool. It solves the biasing problem by minimizing the cost function showed in (6) that relates the circuit performances (β_p(**x**)), the sizing rules (β_c(**x**)) and the design parameters (**x**) with the possibility to include mismatch in the sizing process. In the case of WiCkeD, the sizing problem is solved by using a customized Sequential Quadratic Programming.

$$\min\sum_i exp\left(-a\cdot\left(\beta_{pi}(\mathbf{x})+\beta_{ci}(\mathbf{x})\right)\right),\quad a>0 \tag{6}$$

There are automatic ways to extract DOCs for both CMOS and bipolar circuits [31]. The methodology is based on a library of hierarchically-organized analog building blocks, including single transistors, current mirrors, level shifters, voltage references, differential pairs and more complex cells. The elements of each level have their own sizing rules / DOCs.

7 Conclusion

The OP formulation is how analog designers traditionally approach the sizing and biasing problem in a manual fashion. The OP formulation is also an approach to framing the problem for automated sizing and biasing. Rather than making widths and lengths the independent design variables, the OP formulation instead uses currents and voltages, or other variables that are intuitive to the designer. This transformation has benefits including faster convergence and better quality results. It can be performed with different MOS models and technologies, without a deep insight on the circuit to size. It can be used for sizing cell level circuits, and higher levels as well.

The OP formulation has applications beyond sizing, including classification / regression, symbolic modeling, topology selection / synthesis, portable performance models, new manual sizing flows, and rapid porting.

Despite these benefits and applications, the OP formulation has been underused in the analog CAD literature, most likely due to a lack of familiarity, inconvenience, and underappreciation of its benefits. This paper's main contribution is to expose its diverse benefits, and make its application more familiar and convenient via a thorough literature survey.

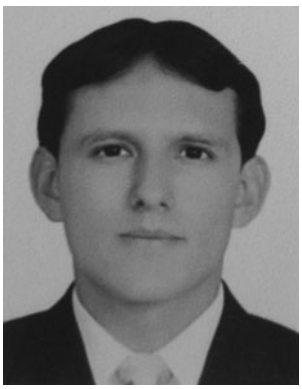
Acknowledgment This work is partially supported by CONACyT-México under project 131839-Y.

References

1. Aggarwal, V. & O'Reilly, U. M. (2006). Design of posynomial models for mosfets: Symbolic regression using genetic algorithms. In: R. L. Riolo, T. Soule, & B. Worzel (Eds.), Genetic programming theory and practice IV, *Genetic and evolutionary computation*, vol. 5, chap. 7. Springer. http://people.csail.mit.edu/unamay/publications-dir/gptp06.pdf.
2. Antreich, K., Eckmueller, J., Graeb, H., Pronath, M., Schenkel, F., Schwencker, R., & Zizala, S. (2000). WiCkeD: Analog circuit synthesis incorporating mismatch. Proceedings of the IEEE Custom Integrated Circuits Conference pp. 511–514.
3. Binkley, D., Hopper, C., Tucker, S., Moss, B., Rochelle, J., & Foty, D. (2003). A CAD methodology for optimizing transistor

current and sizing in analog CMOS design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22*(2), 225–237.
4. Boos, V., Nowak, J., Sylvester, M., Henker, S., Höppner, S., Grimm, H., Krausse, D., & Sommer, R. (2011). Strategies for initial sizing and operating point analysis of analog circuits. Design, Automation and Test in Europe Conference and Exhibition pp. 1–3.
5. Chang, H. (1997). A top-down constraint-driven design methodology for analog integrated circuits. Boston: Kluwer Academic. http://books.google.com/books?id=-C1CzmaxD-wC.
6. Daems, W., Gielen, G., & Sansen, W. (2003). Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transaction on CAD of Integrated Circuits and Systems, 22*(5), 517–534.
7. De Bernardinis, F., Nuzzo, P., & Sangiovanni Vincentelli, A. (2005). Mixed signal design space exploration through analog platforms. Design Automation Conference pp. 875–880.
8. De Bernardinis, F. & Sangiovanni Vincentelli, A. (2005). Efficient analog platform characterization through analog constraint graphs. International Conference on Computer-Aided Design pp. 415–421.
9. De Smedt, B., & Gielen, G. (2003). WATSON: design space boundary exploration and model generation for analog and RFIC design. *Transactions on Computer-Aided Design of Integrated Circuits and Systems, 22*(2), 213–224.
10. Degrauwe, M., Nys, O., Dijkstra, E., Rijmenants, J., Bitz, S., Goffart, B., Vittoz, E., Cserveny, S., Meixenberger, C., van der Stappen, G.,& Oguey, H. (1987). IDAC: an interactive design tool for analog CMOS circuits. *IEEE Journal of Solid-State Circuits, 22*(6), 1106–1116.
11. Ding, M. & Vemuri, R. (2005). A combined feasibility and performance macromodel for analog circuits. Design Automation Conference pp. 63–68.
12. Duarte-Villaseñor, M. A., Tlelo-Cuautle, E., & Gerardo de la Fraga, L. (2011). Binary genetic encoding for the synthesis of mixed-mode circuit topologies. Circuits, Systems and Signal Processing doi:10.1007/s00.034-011-9353-2.
13. Eeckelaert, T., McConaghy, T., & Gielen, G. (2005). Efficient multiobjective synthesis of analog circuits using hierarchical pareto-optimal performance hypersurfaces. *Design, Automation and Test in Europe, 2*, 1070–1075.
14. Flandre, D., Viviani, A., Eggermont, J. P., Gentinne, B., & Jespers, P. (1997). Improved synthesis of gain-boosted regulated-cascode CMOS stages using symbolic analysis and gm/ID methodology. *IEEE Journal of Solid-State Circuits, 32*(7), 1006–1012.
15. Girardi, A., Cortes, F., & Bampi, S. (2006) A tool for automatic design of analog circuits based on g_m/I_D methodology. IEEE International Symposium on Circuits and Systems.
16. Hammouda, S., Dessouky, M., Tawfik, M., & Badawy, W. (2004). A fully automated approach for analog circuit reuse. International Workshop on System-on-Chip for Real-Time Applications, pp. 237–240.
17. Hammouda, S., Said, H., Dessouky, M., Tawfik, M., Nguyen, Q., Badawy, W., Abbas, H., & Shahein, H. (2006). Chameleon art: A non-optimization based analog design migration framework. Proceedings of the Design Automation Conference pp. 885 –888.
18. Hashemian, R. (2012). Application of fixator-norator pairs in designing active loads and current mirrors in analog integrated circuits. IEEE Trans on Very Large Scale Integration (VLSI) Systems doi:10.1109/TVLSI.2011.2172229.
19. Heikkilau, P., Valtonen, M., & Pohjonen, H. (1989). Automated dimensioning of MOS transistors without using topology-specific explicit formulas. *International Symposium on Circuits and Systems, 2*, 1131–1134.
20. Iskander, R., Dessouky, M., Aly, M., Magdy, M., & Hassan, N. (2003). Synthesis of CMOS analog cells using AMIGO. Design, Automation and Test in Europe Conference and Exhibition pp. 297–302.
21. Iskander, R., Galayko, D., Louerat, M. M., & Kaiser, A. (2007). Knowledge-aware synthesis using hierarchical graph-based sizing and biasing. IEEE Northeast Workshop on Circuits and Systems pp. 984–987.
22. Iskander, R., Loüerat, M. M., & Kaiser, A. (2012). Hierarchical sizing and biasing of analog firm intellectual properties. Integration, the VLSI Journal pp. 1–17. doi:10.1016/j.vlsi.2012.01.001.
23. Iskander, R., Loüerat, M. M., & Kaiser, A. (2008). Automatic dc operating point computation and design plan generation for analog IPs. *Analog Integrated Circuits and Signal Processing, 56*(1-2), 93–105.
24. Javid, F., Iskander, R., Durbin, F., & Louerat, M. (2012). Analog circuits sizing using the fixed point iteration algorithm with transistor compact models. IEEE International Conference Mixed Design of Integrated Circuits and Systems pp. 45–50.
25. Javid, F., Iskander, R., & Louerat, M. M. (2009). Simulation-based hierarchical sizing and biasing of analog firm IPs. Behavioral Modeling and Simulation Workshop pp. 43–48.
26. Koh, H., Sequin, C., & Gray, P. (1990). OPASYN: a compiler for CMOS operational amplifiers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 9*(2), 113–125.
27. Kruiskamp, W. & Leenaerts, D. (1995). DARWIN: CMOS opamp synthesis by means of a genetic algorithm. Conference on Design Automation pp. 433–438.
28. Leyn, F., Gielen, G., & Sansen, W. (1998). An efficient DC root solving algorithm with guaranteed convergence for analog integrated CMOS circuits. International Conference on Computer-Aided Design, pp. 304–307.
29. Liu, B., Pak, M., Zheng, X., & Gielen, G. (2011). A novel operating-point driven method for the sizing of analog IC. IEEE International Symposium on Circuits and Systems, pp. 781–784.
30. del Mar Hershenson, M., Boyd, S., & Lee, T. (1998). GPCAD: A tool for CMOS op-amp synthesis. Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design, pp. 296–303. ACM http://doi.acm.org/10.1145/288548.288628.
31. Massier, T., Graeb, H., & Schlichtmann, U. (2008). The sizing rules method for CMOS and bipolar analog integrated circuit synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27*(12), 2209–2222.
32. Maulik, P., Carley, L., & Rutenbar, R. (1995). Integer programming based topology selection of cell-level analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14*(4), 401–412.
33. McConaghy, T. & Gielen, G. (2009). CAFFEINE: template-free symbolic model generation of analog circuits via canonical form functions and genetic programming. *Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28*(8), 1162–1175.
34. McConaghy, T., Palmers, P., Steyaert, M., & Gielen, G. (2009). Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28*(9), 1281–1294.
35. Medeiro, F., Fernández, F., Domínguez-Castro, R., & Rodríguez-Vázquez, A. (1994) A statistical optimization-based approach for automated sizing of analog cells. International Conference on Computer-Aided Design pp. 594–597.
36. Ochotta, E., Rutenbar, R., & Carley, L. (1996). Synthesis of high-performance analog circuits in ASTRWOBLX. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15*(3), 273–294.

37. Phelps, R., Krasnicki, M., Rutenbar, R., Carley, L., & Hellums, J. (2000). Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 19*(6), 703–717.
38. Sansen, W. (2006). *Analog design essentials*. New York: Springer.
39. Savio, A., Colalongo, L., Kovacs-Vajna, Z., & Quarantelli, M. (2004). Scaling rules and parameter tuning procedure for analog design reuse in technology migration. International Symposium on Circuits and Systems, vol. 5, pp. 117–120
40. Savio, A., Colalongo, L., Quarantelli, M., & Kovacs-Vajna, Z. M. (2006). Automatic scaling procedures for analog design reuse. *IEEE Transactions on Circuits and Systems I: Regular Papers, 53*(12), 2539–2547.
41. Sheinman, B., Ramm, D., & Nachmany, D. (2008). A comprehensive flow for transfer of designs between technologies. Proceedings of the International Conference on Microwaves, Communications, Antennas and Electronic Systems, pp. 1–7.
42. Sheu, B., Fung, A., & Lai, Y. N. (1988). A knowledge-based approach to analog IC design. *Transactions on Circuits and Systems, 35*(2), 256–258.
43. Silveira, F., Flandre, D., & Jespers, P. (1996). A g_m/I_D Based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA. *IEEE Journal of Solid-State Circuits, 31*(9), 1314–1319.
44. Stefanovic, D., Kayal, M., & Pastre, M. (2005). PAD: A New Interactive Knowledge-Based Analog Design Approach. *Analog Integrated Circuits and Signal Processing, 42*(3), 291–299.
45. Tadeusiewicz, M., & Halgas, S. (2011). A contraction method for locating all the dc solutions of circuits containing bipolar transistors. Circuits, Systems and Signal Processing pp. doi: 10.1007/s00.034-011-9362-1.
46. Torres-Munoz, D., & Tlelo-Cuautle, E. (2005). Automatic biasing and sizing of CMOS analog integrated circuits. Midwest Symposium on Circuits and Systems, 1, 915–918.
47. Toumazou, C., Makris, C., & Berrah, C. (1990). ISAID-a methodology for automated analog IC design. International Symposium on Circuits and Systems, 1, 531–535.
48. Yang, X., Xin, L., Kan-Lin, H., Boyd, S., & Nausieda, I. (2005). OPERA: Optimization with ellipsoidal uncertainty for robust analog IC design. Design Automation Conference pp. 632–637.



Ivick Guerra-Gómez received a B.Sc. degree from Benemerita Universidad Autonoma de Puebla (BUAP), Mexico in 2006, honored with cum laude, and M.Sc. and Ph.D. degrees from Instituto Nacional de Astrofisica, Optica y Electronica (INAOE), Mexico in 2008 and 2012, respectively. He has authored about 20 articles published in book chapters, journals and conferences. His research interests include integrated circuit design and optimization, evolutionary algorithms and analog CAD.

Trent McConaghy has Bachelors in Engineering and in Computer Science (both with distinction) from the University of Saskatchewan, and received its 2001 Outstanding Young Alumni Award for significant accomplishments since graduation. He got his Ph.D. from Katholieke Universiteit Leuven, Belgium in 2008, his doctoral thesis won the international EDAA Outstanding Dissertation Award. He is co-founder and Chief Scientific Officer of Solido Design Automation Inc. He was a co-founder and Chief Scientist of Analog Design Automation Inc., which was acquired by Synopsys Inc. in 2004. He did research for the Canadian Department of National Defense. He has authored approximately 30 journal papers, book chapters, and conference papers, and has about 20 patents granted / pending. He is author of the book"Variation-Aware Analog Structural Synthesis: A Computational Intelligence Approach" (Springer, 2009), and the book "Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide" (Springer, 2012). He has been an invited speaker at many labs, universities, and conferences ranging from JPL to MIT. He is regularly a technical program committee member and reviewer in both the CAD and intelligent systems fields, such as IEEE Trans CAD, ACM TODAES, Electronics Letters, to the Journal of Genetic Programming and Evolvable Hardware, GECCO, etc. His research interests include: statistical machine learning and intelligent systems, with transistor-level CAD applications such as statistically-aware design, analog topology design, automated sizing, knowledge extraction, and symbolic modeling.

Esteban Tlelo-Cuautle received a B.Sc. degree from Instituto Tecnologico de Puebla (ITP), Mexico in 1993. He then received both M.Sc. and Ph.D. degrees from Instituto Nacional de Astrofisica, Optica y Electronica (INAOE), Mexico, in 1995 and 2000, respectively. He is appointed as Professor-Researcher at INAOE. He is an IEEE Senior Member and member of the editorial board in several journals. He regularly serves as a reviewer in more than 20 high impact-factor journals, and 20 internationally recognized conferences. He has been a member of Program Committees in several prestigious international conferences, such as IEEE LASCAS, ICECS and SMACD. His research interests include systematic synthesis and behavioral modeling and simulation of linear and nonlinear circuits and systems, chaotic oscillators, symbolic analysis, multi-objective evolutionary algorithms, and analog/RF and mixed-signal design automation tools.