A Tool for Fast Mismatch Analysis of Analog Circuits

R. Rodríguez-Macías, F.V. Fernández, A. Rodríguez-Vázquez, and J.L. Huertas
Dept. of Analog Circuit Design, Centro Nacional de Microelectrónica.
Edificio CICA, Avda. Reina Mercedes s/n, 41012-Sevilla, Spain.

ABSTRACT

A tool is presented that evaluates statistical deviations in performance characteristics of analog circuits, starting from statistical deviations in the technological parameters of MOS transistors. Performance is demonstrated via the analysis of a Miller OTA in two different configurations and a linearized CMOS transconductor. The CPU time is reduced by a factor of 25 to 90 with respect to conventional Monte Carlo simulation, while maintaining similar accuracy in the computations.

I. Introduction

The performance of analog cells may be severely deteriorated due to random fluctuations in the electrical parameters of their components: transistors, resistors and capacitors. Crucial features like offset, common-mode rejection ratio (CMRR), power-supply rejection ratio (PSRR), etc., are critically dependent on random-induced mismatches, and require careful design (equivalently sizing) to guarantee that specifications remain within tolerable margins for the largest possible number of circuit samples. However, due to the difficulties encountered in the modeling and evaluation of random fluctuations, this problem is usually handled by simply imposing conservative constraints in the design parameter space; e.g. the transistor area in a differential amplifier must be larger than say, 100 \( \mu \text{m}^2 \). This leads to underexploitation of the performance potentials of the technologies. The optimization of the design procedure of analog cells under random fluctuations of the transistor parameters encompasses two different kinds of problems:

a) **Statistical modeling**, essentially the capture of statistical features of the components into parameterized models.

b) **Statistical simulation**, or the evaluation of the statistical features of the circuit performance induced by random variations of the electrical parameters.

The statistical modeling of MOS components has been covered by different authors in recent years [1-4]. Our contribution focuses not on the statistical modeling, but rather on statistical simulation. For modeling we rely on the proposal by Pelgrom [1] and more specifically, its further refinement as proposed in [2]. Such a model provides the variances of the electrical parameters of MOS devices as a function of their geometry and distances and covers correlations among electrical parameters.

According to [2] variances of the electrical parameters of a given device can be evaluated from its layout as

\[
\sigma^2 (\Delta P) = \frac{\Delta_p}{2WL} + \frac{\Delta_p^2}{2W^2L^2} \tag{1}
\]

where \( D \) denotes the distance from the device to a common reference point (the coordinate center); \( WL \), its area; and \( \Delta_p \) and \( \Delta_p \) are fitting constants, different for each technological parameter subject to variation.

Let us focus on statistical simulation. Most analog designers use Monte Carlo analysis for this purpose. However, for large analog cells described at the electrical level this is very costly in CPU time. This precludes its usage inside iterative analog design procedures [5,6] and motivates looking for more efficient solutions [7,8,9]. Especially significant is the technique proposed by [9], based on a two-step linearization of the circuit performance. Precise results are obtained using the Monte Carlo method with that linearization. However, it assumes that circuit performances do not change significantly. Thus, the examples presented have no node with voltage standard deviation larger than 0.1V.

Our contribution aims to use matrix methods to perform statistical analysis of analog integrated circuits starting from correlated electrical model parameters. These have been calculated according to the modeling technique in [4]. In particular, two different matrix techniques have been developed and included in a computer tool whose flowchart is shown in Fig.1. Some significant features of the proposed method are:

1) Statistical features of branch currents and node voltages of any circuit can be evaluated, allowing easy extension to any electrical, DC or AC, performance.

2) Results adequately fit those obtained with Monte Carlo analysis using an electrical simulator.

3) Analysis time is minimized so that the method can be included in an optimization-based design system.

II. Matrix-I Technique

Variations in the transistor electrical parameters of a
given circuit induce changes in its node voltages and branch currents which can be calculated analytically.

For a single transistor:
\[
\sum \frac{\partial i}{\partial p_k} \Delta V_i + \frac{\partial i}{\partial V_G} \Delta V_G + \frac{\partial i}{\partial V_D} \Delta V_D + \frac{\partial i}{\partial V_S} \Delta V_S + \frac{\partial i}{\partial V_B} \Delta V_B = \Delta i
\]

where \( p_k \) represents the \( k \)-th electrical model parameter.

Equation (2) is more conveniently written as,
\[
\sum \frac{\partial i}{\partial p_k} A_k = \Delta i
\]

For a given circuit containing \( N \) transistors, and assuming that \( N \) node voltage variations and \( N-M \) branch current variations are chosen as unknowns, the following linear equation system can be formulated:
\[
Ax = b
\]

where
\[
\Delta x = \Delta x = \begin{bmatrix} \Delta V_1 \\ \vdots \\ \Delta V_N \end{bmatrix}
\]
\[
b = \begin{bmatrix} -A_v \\ \vdots \\ -A_v \end{bmatrix}
\]

(4)

Cramer’s rule is used to solve (4) and enables introducing correlations between transistors in the calculation. Hence, the \( j \)-th unknown is given by:
\[
\Delta x_j = \frac{\Delta x_j}{A_{j,j} - \sum_{k=1}^{N} A_{j,k} \Delta x_k}
\]

(6)

where
\[
\Delta x_j = \begin{bmatrix} \Delta V_1 \\ \vdots \\ \Delta V_N \end{bmatrix}
\]
\[
b_j = -\sum_{k=1}^{N} A_{j,k} \Delta x_k
\]

(5)

\[ A_{j,j} = \begin{cases} \frac{\partial i}{\partial V_j} & \text{if } j \leq M \\ \frac{\partial i}{\partial V_D} - \frac{\partial i}{\partial V_G} + \frac{\partial i}{\partial V_S} & \text{if } j > M \end{cases} \]

For another vector of electrical parameters \( p_i \) transistor currents can be approximated as:
\[
\Delta i = \sum_{k=1}^{N} \frac{\partial i}{\partial p_k} \Delta p_k
\]

(9)

\[ p_{i,j} = \begin{cases} 1 & \text{if } k = i \\ \frac{\Delta i}{\sum_{k=1}^{N} \frac{\partial i}{\partial p_k} \Delta p_k} & \text{otherwise} \end{cases}
\]

Table 1 shows the deviations in node voltages and branch currents obtained by Monte Carlo simulations (50 simulation instances) and those obtained by the Matrix-1 technique. As shown, the Matrix-1 technique reduces the CPU time by a factor of 90, without significant deviations in the analysis results as compared with those obtained by Monte Carlo simulations. The BSIM model was used considering variations of 16 parameters (and their correlations). Correlations of the electrical parameters in HSPICE simulations were calculated using the Principal Component Analysis technique [2].

However, the Matrix-1 method does not consider eventual excursions of the transistor operating points through different operating regions (i.e. ohmic, saturation, etc.) and thus, may obtain inaccurate results for circuits containing nodes which swing over wide voltage intervals. For example, consider the open loop OTA Miller of Fig.3. Table 2 shows the branch currents and node voltages obtained by Monte Carlo analysis and the Matrix-1 technique. Since the circuit is in open loop, the voltage \( V_{n3} \) changes over a wide range, taking transistors \( M_6 \) and \( M_7 \) out of their nominal operating regions and hence, introducing important analysis errors.

### III. Monte Carlo analysis on the system matrix: Matrix-2 Technique

The Matrix-2 technique solves these problems by performing a Monte-Carlo analysis on vector \( b \). Parameters are generated according to their distribution. A non-linear system is solved using the Newton-Raphson method and sensitivities are updated at each iteration.

Let us define a function \( F_i \) for each transistor:
\[
F_i = I_i - I_i(V,p_i) 
\]

(10)

Circuit analysis consists in determining the voltages and currents which annul all \( F_i \). A Monte Carlo simulation consists in solving the system for a large number of vectors of electrical parameters \( p_i \).

The time is reduced considerably by using the system solution at the nominal point as an initial solution of the Newton-Raphson method. The function \( F_i \) at the nominal point is:
\[
F_i = I_i(V,p) - I_i(V,p_0) 
\]

(11)

For another vector of electrical parameters \( p_i \) transistor currents can be approximated as:
\[
I_i = I_i(V,p_i) = I_i(V,p_0) + \sum \frac{\partial I_i}{\partial p_k} \Delta p_k
\]

(12)

where the derivative is evaluated at the nominal point. Its corresponding function \( F_i \) is then:
\[
F_i = I_i(V,p_i) - I_i(V,p_0) - \sum \frac{\partial I_i}{\partial p_k} \Delta p_k
\]

(13)

Let us define a function \( G_i \) as the difference of \( F_i \) at the nominal point and \( F_i \) for another set of parameters \( p_i \).
Expanding $G_i$ in Taylor series (first order) and making it equal to zero to apply the Newton-Raphson method:

$$G_i(x) + \delta x = 0 \quad (15)$$

Applying (15) to (14) and reordering terms obtains in matrix notation:

$$A \Delta x = \delta$$

where $A$ is defined as in (5) and $\Delta x$ is the vector of unknown variations in voltages and currents.

In every iteration of the Newton-Raphson method, the equation system is solved by LU decomposition and $Ax$ is updated summing the solution $(Ax)$ to it. The iterative procedure continues until convergence.

IV. Statistical features of DC Characteristics

Interesting DC characteristics are functions of node voltages and branch currents at one or more operating points. As their relationships are known, their statistical features are easily obtained starting from the voltage and current deviations at the operating points needed for their evaluation. The variation in a DC performance $Y$, is expressed as:

$$Y_i = Y_i(\Delta x_1, \ldots, \Delta x_m; \Delta x_1, \ldots, \Delta x_m) \quad (17)$$

where $\Delta x_k$ is the increase in the $k$-th unknown at the $j$-th operating point. Performances are usually determined by only one unknown; so, (17) is simplified to:

$$Y_i = Y_i(\Delta x_1, \ldots, \Delta x_m) \quad (18)$$

Method-1 calculates performance sensitivities with respect to vector $\Delta X$, and the variance in then given by:

$$\sigma^2(\Delta P) = \sum \left( \frac{\partial P}{\partial x} \right)^2 \sigma^2 \Delta x + 2 \sum \sum \frac{\partial P}{\partial x} \frac{\partial P}{\partial x} \Delta x_k \Delta x_m$$

where

$$\Delta x_k \cdot \Delta x_m = \sum \sum A_{km} P_{k} \sigma_{n} \sigma_{m}$$

and $P_{k}$ is defined similarly to (9).

If Method-2 is used, calculations are reduced to using conventional statistical estimators.

Table 3 shows experimental results for the linearized transconductor in Fig.4. The specifications considered are slope, linearity, input offset, and output offset. Methods 1 and 2 give similar results (except mean values which are assumed equal to the nominal ones in Method-1), because no node experiments large excursions. Speed with respect to Monte Carlo analysis is increased by a factor of around 25 with Method-2 and a factor of 80 using Method-1.

V. References

Table 1: Comparative results for Fig. 2.

<table>
<thead>
<tr>
<th>Node voltages &amp; branch currents</th>
<th>Monte Carlo (50 samples)</th>
<th>Matrix-1 technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>std dev</td>
<td>std dev</td>
</tr>
<tr>
<td>$V_{n1}$</td>
<td>5.29mV</td>
<td>4.39mV</td>
</tr>
<tr>
<td>$V_{n2}$</td>
<td>6.24mV</td>
<td>5.26mV</td>
</tr>
<tr>
<td>$V_{n3}$</td>
<td>4.94mV</td>
<td>3.15mV</td>
</tr>
<tr>
<td>$V_{n4}$</td>
<td>4.23mV</td>
<td>3.46mV</td>
</tr>
<tr>
<td>$V_{n5}$</td>
<td>7.94mV</td>
<td>7.43mV</td>
</tr>
<tr>
<td>$I_{br1}$</td>
<td>0.22µA</td>
<td>0.22µA</td>
</tr>
<tr>
<td>$I_{br2}$</td>
<td>0.23µA</td>
<td>0.22µA</td>
</tr>
<tr>
<td>$I_{br3}$</td>
<td>1.28µA</td>
<td>1.18µA</td>
</tr>
</tbody>
</table>

CPU time: 90s

Table 2: Comparative results for the circuit in Fig. 3.

<table>
<thead>
<tr>
<th>Node voltages &amp; branch currents</th>
<th>Monte Carlo analysis (50 samples)</th>
<th>Matrix-1 technique</th>
<th>Matrix-2 technique (50 samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>mean std dev</td>
<td>std dev</td>
<td>mean std dev</td>
</tr>
<tr>
<td>$V_{n1}$</td>
<td>-3.89V 5.29mV</td>
<td>4.39mV</td>
<td>-3.89V 4.95mV</td>
</tr>
<tr>
<td>$V_{n2}$</td>
<td>-1.27V 3.5mV</td>
<td>2.72mV</td>
<td>-1.27V 3.21mV</td>
</tr>
<tr>
<td>$V_{n3}$</td>
<td>-3.04V 3.11V</td>
<td>0.4V</td>
<td>-3.13V 1.18V</td>
</tr>
<tr>
<td>$V_{n4}$</td>
<td>4.11V 4.39mV</td>
<td>3.34mV</td>
<td>4.11V 3.86mV</td>
</tr>
<tr>
<td>$V_{n5}$</td>
<td>4.17V 0.31V</td>
<td>0.2V</td>
<td>3.91V 0.33V</td>
</tr>
<tr>
<td>$I_{br1}$</td>
<td>7.27µA 0.27µA</td>
<td>0.27µA</td>
<td>7.14µA 0.27µA</td>
</tr>
<tr>
<td>$I_{br2}$</td>
<td>7.24µA 0.29µA</td>
<td>0.27µA</td>
<td>7.28µA 0.29µA</td>
</tr>
<tr>
<td>$I_{br3}$</td>
<td>27.28µA 33.9µA</td>
<td>35.7µA</td>
<td>32.2µA 35.2µA</td>
</tr>
</tbody>
</table>

CPU time: 50s

Table 3: Comparative results for the circuit in Fig. 4 using Monte Carlo simulation and the new tool.

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Monte-Carlo</th>
<th>New Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean value</td>
<td>standard deviation</td>
</tr>
<tr>
<td>Slope (Vgs=0)</td>
<td>5.5 µA/V</td>
<td>0.21µA/V</td>
</tr>
<tr>
<td>Output offset</td>
<td>-0.1nA</td>
<td>0.5nA</td>
</tr>
<tr>
<td>Input offset</td>
<td>-0.4mV</td>
<td>0.1V</td>
</tr>
<tr>
<td>Linearity: V1H(V)=0.9I(-2.5)</td>
<td>-1.1V</td>
<td>20mV</td>
</tr>
<tr>
<td>CPU time</td>
<td>800s</td>
<td>Matrix-1 10s / Matrix-2 30s</td>
</tr>
</tbody>
</table>

Figure 2: Unity-gain Miller OTA

Figure 3: Open-loop Miller OTA

Figure 4: Linearized CMOS transconductor