Automated Analog IC Design Methodology for a Large Number of Corner Points

Árpád Bűrmen, Janez Puhan, Iztok Fajfar, and Tadej Tuma
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, 1001 Ljubljana, Slovenia
arpadb@fides.fe.uni-lj.si

Abstract

Robust analog ICs fulfill the design requirements for a range of operating conditions and manufacturing process variations. A combination of a particular manufacturing process variation and values of operating conditions is also known as a corner point. It is common practice in the design phase to evaluate the circuit only for corner points that represent extreme variations of the manufacturing process and operating conditions. But even this simplification means that every candidate circuit must be evaluated several ten or even several hundred times, making automated robust design virtually impossible. A simple approach utilizing multiple optimization runs where corner points are gradually added to the set of examined corner points is presented and illustrated with a sample automated differential amplifier design. The results show that the final result is obtained in much shorter time than it would be obtained in a single run including all corner points.

1 Introduction

Automated robust design of analog ICs [1] is becoming more and more important as the market requires from IC designers to obtain solutions faster. One of the main obstacles in automating this process is the fact that the computer must evaluate a large number of circuits in order to come up with a feasible solution. The main guide for a designer is the set of design requirements (e.g. gain above 60dB, bandwidth above 10MHz, area below 1000μm, ...). A solution is considered to be robust if the design requirements are fulfilled not only for the nominal values, but for a given range of process variations (e.g. worst power, worst zero, etc.) and operating conditions (e.g. temperatures between -50°C and 150°C, supply voltage between 1.6V and 2.0V, ...).

This requirement is usually relaxed in the design process by making sure the circuit fulfills the design requirements only for extreme combinations of process variations and operating conditions (corner points). But even by resorting to this simplification the number of corner points can be quite large. Take for instance a design that must fulfill the design requirements at 4 extreme process variations, temperature range (2 extreme values), supply voltage range (2 extreme values), and load resistance (2 extreme values). The total number of corner points in this case would be 32. This means that every candidate circuit must be evaluated (simulated) 32 times before one can say anything about its robustness. If we consider that automated design by means of optimization evaluates several thousand candidate circuits before it finds a solution we quickly arrive at the conclusion that the process of automated design would simply take too long.

We propose an approach where the circuit is designed in multiple optimization runs. In a single run only a subset of the complete set of corner points is accounted for. After every such run the resulting circuit is evaluated in all corner points (full evaluation). The next run accounts for a few corners more, depending on the outcome of the full evaluation performed after the previous run.

The remainder of this paper is organized as follows. First the robust design problem is formulated mathematically. The multiple run strategy in described. Finally a sample design of a differential amplifier is presented and the results discussed.

2 Robust design

The set of corner models describes the extreme values of process variations. We denote this set $\mathcal{C} = \{c_1, c_2, ..., c_{m_0}\}$. Let $e_i,\ i = 1, 2, ..., r$ denote the r parameters that specify the operating condition (e.g. temperature, supply voltage, ...). Every operating condition parameter has a set of values $\mathcal{E}_i = \{e_i^1, e_i^2, ..., e_i^{m_r}\}$ that are of some interest to the designer. Usually such a set comprises the nominal, minimal, and maximal value of the parameter.

A corner point is a $r+1$-tuple where the first component is a corner model and the remaining r components are values of operating condition parameters. There is a total of $\prod_{i=0}^{r} m_i$ corner points defined by the set of corner models and available values of operating condition parameters. Let $\mathcal{U}_{\text{total}}$ denote the set of all corner points. On the other hand we denote the se of all corner points that are of some interest to the designer by $\mathcal{U}_0$. Of course $\mathcal{U}_0 \subseteq \mathcal{U}_{\text{total}}$.

The performance of the circuit is described by circuit characteristics $y_i \in \mathbb{R}$ (e.g. gain, bandwidth, area, ...). We restrict ourselves to sizing circuits with predefined structure. Only the values of circuit parameters $x_i,\ i = 1, 2, ..., N$ (e.g. widths and lengths of MOS transistors, values of resistors and capacitors, ...) are allowed to change.

The circuit characteristics depend on the values of the circuit parameters and on the corner point for which the circuit is evaluated. We denote this by $y_i(\mathbf{x}, \mathbf{p})$ where $\mathbf{x} = [x_1, x_2, ..., x_N]$ and $\mathbf{p} \in \mathcal{U}_0$.

The goals that must be achieved by the design process are specified by means of design requirements expressed as lower (b_i) and upper (B_i) bound on circuit characteristics. Usually one doesn't want to bound a circuit characteristic from above and below. Take for instance gain. In order to require gain above 60dB one must set b to 60 and B to ∞. On the other hand the requirement that delay should be below 10ms implies $b = -\infty$ and $B = 0.001$.

A circuit corresponding to the parameter vector $\mathbf{x}$ is considered to be robust across the set of corner points $\mathcal{U} \subseteq \mathcal{U}_0$ if the following relations hold

$$b_i \leq y_i(\mathbf{x}, \mathbf{p}) \leq B_i \quad i = 1, 2, ..., n \quad \forall \mathbf{p} \in \mathcal{U} \tag{1}$$

Requirements (1) can be transformed into a scalar cost function (CF) $f_{\mathcal{U}}(\mathbf{x})$ [1] for which $f_{\mathcal{U}}(\mathbf{x}) = 0$ holds if the circuit is robust across the set of corner points $\mathcal{U}$ and $f_{\mathcal{U}}(\mathbf{x}) \neq 0$ otherwise. This way the search for the set of circuit parameters $\mathbf{x}$ that results in a robust circuit is transformed into the search for $f_{\mathcal{U}}(\mathbf{x}) = 0$. In practice such search can be implemented by means of optimization algorithms (e.g. [2]).

In practice the circuit must be evaluated once for every member of $\mathcal{U}$ in order to obtain the value of $f_{\mathcal{U}}(\mathbf{x})$. As the number of corner points that are being considered in the process of robust design can be quite large, the evaluation of $f_{\mathcal{U}}(\mathbf{x})$ can take a significant amount of time.

An optimization run starts with an initial point $\mathbf{x}_0$ in the circuit parameter space and searches for $\mathbf{x}$ where $f_{\mathcal{U}}(\mathbf{x})$ is as low as possible or in other words searches for a circuit that satisfies all design requirements in all corner poitns from $\mathcal{U}$.

After every optimization run the resulting circuit is checked across all corners points. Particularly interesting are those circuit characteristics that fail to fulfill the design requirement in at least one corner point from $\mathcal{U}_0$. Let $\mathcal{A}$ denote the set of indices corresponding to these circuit characteristics (failed circuit characteristic indices).

For every circuit characteristic y_i that fails to fulfill the design requirement there is a corner where y_i reaches its worst value. Formally this corner can be expressed as

$$\mathbf{q}_i = \arg\max_{\mathbf{p} \in \mathcal{U}_0} \max(y_i(\mathbf{x}, \mathbf{p}) - B_i, b_i - y_i(\mathbf{x}, \mathbf{p})) \tag{2}$$

Let $\mathcal{B}$ denote the set of corners that correspond to the worst values of circuit characteristics that fail to fulfill the design requirements.

$$\mathcal{B} = \{\mathbf{q}_i : i \in \mathcal{A}\} \tag{3}$$

Using the notation presented above we can formulate the algorithm that gradually increases the number of corner points accounted for in subsequent optimization runs, until a circuit satisfying the design requirements across all corner points from $\mathcal{U}_0$ is found.

3 The Algorithm

The algorithm starts with an initial set of corner points $\mathcal{U}_1$. Usually this set contains only the nominal corner point.

1. Set $i := 1$.
2. Choose an initial set of corners $\mathcal{U}_1$.
3. Choose an initial point $\mathbf{x}_{i-1}$ in the circuit parameter space.
4. Minimize $f_{\mathcal{U}_i}(\mathbf{x})$, use $\mathbf{x}_{i-1}$ as initial point.
5. Evaluate the resulting circuit across all corner points from $\mathcal{U}_0$.
6. Form the set of failed circuit characteristic indices ($\mathcal{A}$) and the corresponding set of corner points ($\mathcal{B}$).
7. Let $\mathcal{U}_{i+1} := \mathcal{U}_i \cup \mathcal{B}$.
8. Let $i := i + 1$.
9. If $i \leq i_{\max}$ and $\mathcal{B} \neq \emptyset$ go back to step 3.

The algorithm finishes after at most $i_{\max}$ optimization runs. Every optimization run is followed by an evaluation of the resulting circuit across the complete set of corner points $\mathcal{U}_0$. In case a circuit is found that after evaluation results in an empty set $\mathcal{B}$, the algorithm has found a solution to the design problem.

The proposed algorithm has one major advantage when compared to a single optimization run that accounts for all corner points in $\mathcal{U}_0$. It attempts to

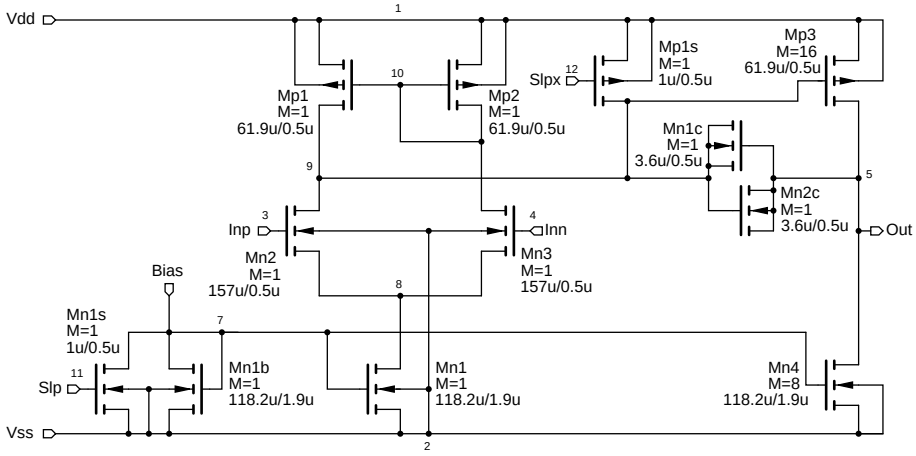

Figure 1: A simple amplifier.

pinpoint the n corner points corresponding to worst values of individual circuit characteristics. Usually these corner points remain unchanged when $\mathbf{x}$ moves in the vicinity of a solution. The vicinity of a solution is often found in the first optimization run when only a small number of corner points is accounted for (usually only the nominal corner point). Therefore one can expect to obtain a circuit that satisfies all design requirements by using only a small subset of $\mathcal{U}_0$ in the optimization runs. This subset is determined automatically by the algorithm.

4 Example

The algorithm is demonstrated on a sample design of a differential amplifier. The topology of the amplifier is depicted in figure 1. The corresponding testbench circuit is in figure 2.

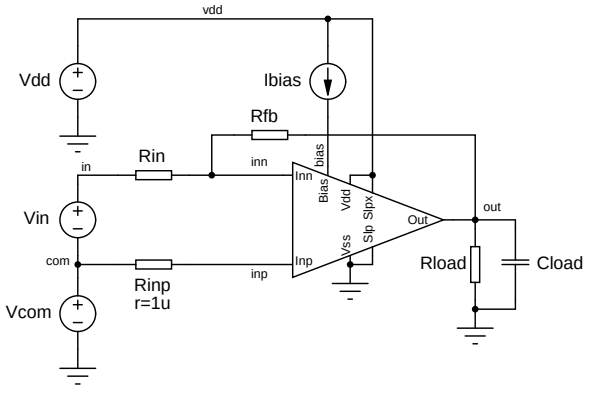

Figure 2: Testbench circuit.

The goal was to design a robust circuit with respect to 2 corner models (FF and SS), temperature range $0°\mathrm{C} - 125°\mathrm{C}$, supply voltage range $2.7\mathrm{V} - 3.3\mathrm{V}$, and bias current range $8\mu A - 12\mu A$. These requirements define 16 extreme corner points ($p_1 - p_{16}$). Beside the extreme corner points a nominal corner point p_0 (corner model TT, 27°C, 3.0V, and 10μA) was also used in the process of automated design.

6 circuit characteristics were subject to optimization: area (goal $< 5000\mu\mathrm{m}^2$), DC supply current (goal $< 150\mu$A), DC swing where differential gain is above 50% of its maximal value (goal $>$ 2.0V), small signal differential gain (goal $>$ 60dB), unity gain-bandwidth (goal $>$ 15MHz), and phase margin (goal $> 45°$).

The characteristics were measured in a closed-loop testbench circuit (figure 2). The circuit characteristics were evaluated from the ratio of amplifier output (node out) to amplifier input (nodes inp and inn). Such a testbench circuit compensates for DC offset voltage and eliminates the need for a very fine DC sweep across a wide input voltage range. The following values were chosen for the testbench circuit: $\mathrm{R_{in}} = \mathrm{R_{fb}} = 100\mathrm{M}\Omega$, $\mathrm{R_{load}} = 10\mathrm{M}\Omega$, $\mathrm{C_{load}} = 2\mathrm{pF}$, $\mathrm{V_{com}} = 1.5\mathrm{V}$.

To reduce the number of circuit parameters subject to optimization, the following matching information was specified (transistors with same width and length: (Mn1b, Mn1, and Mn4), (Mn2, and Mn3), and (Mp1, Mp2, and Mp3). The width of Mn1c and Mn2c was identical and also subject to optimization. Mn1s and Mp1s act as switches and are left unchanged in the process of design. The initial point was chosen to be $5\mu\mathrm{m}/0.5\mu\mathrm{m}$ for all transistors subject to optimization.

Another requirement imposed on the circuit was that $V_{GS} - V_T$ and $V_{DS} - V_{DSAT}$ at the operating point remain positive for all transistors subject to optimization, except for Mn1c and Mn2c.

In the first run $\mathcal{U}_1 = \{p_0\}$, whereas in all subsequent runs the nominal corner point p_0 was not accounted for in the optimization. It was however included in the check performed after every run. This

i	n_{corners}	n_{eval}	t_{run} [s]	n_{failed}	t_{check} [s]
1	1	82	31.0	1	6.0
2	1	144	53.2	2	6.0
3	3	284	311.9	1	6.1
4	4	322	470.0	1	6.0
5	5	242	446.2	0	6.1

Table 1: Run summary. n_{corners}, n_{eval}, t_{run}, n_{failed}, and n_{checked} represent the number of corner points in $\mathcal{U}_i$, the number of CF evaluations, the optimization runtime, the number of corners in $\mathcal{B}$, and the time spent for evaluating the resulting circuit, respectively.

can be justified by the fact that the the circuit's performance is worst at extreme corners and thus there is no need to optimize across the typical corner point. The circuit that was obtained as a result of the i-th run was used as the initial point for the next optimization run.

The algorithm was implemented and SPICE OPUS was used as the circuit simulator [3]. Table 1 summarizes the 5 optimization runs needed to obtain a solution that satisfies all design requirements in all 17 corner points (16 extreme and one nominal).

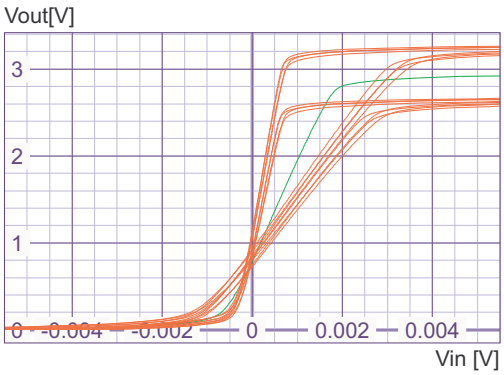

Figure 3: Initial DC response of the amplifier.

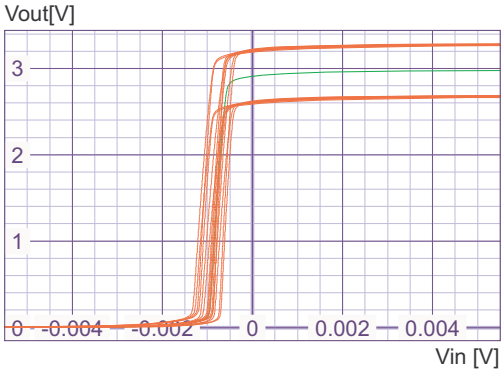

Figure 4: DC response after automated design.

The total time needed for these 5 runs was 1343s. One cannot simply add up the CF evaluations from different runs, as an evaluation across 2 corner points takes twice the time of an evaluation across one corner point. It is more appropriate to add up the number of corner point evaluations. In 5 runs from table 1 a total of 3661 corner point evaluations took place.

On the other hand if only one run is performed with 16 extreme corner points, it takes 6213s to find a solution. The CF is evaluated 572 times. Since 16 corner points are accounted for in the process of optimization it takes a total of 9152 corner point evaluations. This clearly shows the advantage of the proposed approach.

The DC response of the initial circuit across all 17 corner points is depicted in figure 3 along with that of the the final circuit in figure 4. The graphs show that automated design makes sense. The final circuit's performance is better and varies less across corner points.

5 Conclusion

An algorithm for optimizing analog circuits across a large number of corner points was presented. The main idea of the approach is to gradually increase the number of corner points used in successive optimization runs. Its efficiency was demonstrated on a design example of a differential amplifier. The comparison with a single run including all corner points showed that the proposed approach is significantly faster.

6 Acknowledgement

The research was co-funded by the Slovenian Research Agency (Agencija za raziskovalno dejavnost) through the programme P2-0246 Algorithms and optimization methods in telecommunications.

References

[1] Á. Bűrmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar, and T. Tuma. Automated Robust Design and Optimization of Integrated Circuits by Means of Penalty Functions. International Journal of Electronics and Communications, 57: 47–56, 2003.

[2] M. J. Box. A new method of constrained optimization and comparison with other methods. The Computer Journal, 7:42–54, 1965.

[3] J. Puhan and T. Tuma. Optimization of analog circuits with SPICE 3f4. Proceedings of the ECCTD'97, vol. 1, 177–180, 1997.