Improved Design Centering In a Reduced Search Space for Electronic Circuits Optimization

Galia I. Marinova[1], Vassil G. Guliashki[2]

Abstract - **The paper presents an improved version of DCSDR method for efficient design centering for electronic circuits. The choice of simplex vertex for current reflection is more precise. A new step is added for the case when the last iteration was unsuccessful but there are element values, which are not reflected towards the corresponding best value in the simplex, i. e. there are available unexplored search directions. Thus designs which initial values are close to the optimal can also be treated. An extended investigation on different circuits is performed.**[+]

Keywords - **Statistical design methodology, Design centering, Discrete optimization**

I. INTRODUCTION

The design centering objective is to find out the optimal values for the parameters of circuit elements in order to obtain maximal yield. The optimal tolerancing goal is to obtain the optimal tolerance values for the parameters of the circuit elements in order to avoid fail by reducing tolerances where necessary and in order to reduce cost by increasing tolerances where it is possible. Statistical design methodology is based on nominal design and includes design centering and optimal tolerancing. Theory for design centering and optimal tolerancing can be found in [7]. Implementations of design centering are presented in [2, 4]. In design centering a Monte Carlo analysis is performed (by means of IESD statistical simulator [3]) to estimate the yield during the optimization procedure.

The design centering problem can be defined as:

$$\textbf{Min} \qquad F = \sum_{i=1}^{m} \delta_i \tag{1}$$

$$\textbf{subject to:} \quad \delta_i = \begin{cases} 1 \text{ if } J(x) \notin [Lb, Ub]; \\ 0 \text{ if } J(x) \in [Lb, Ub]; \end{cases} \tag{2}$$

$$l_j \le x_j \le u_j; \quad j = 1,\ldots,n; \tag{3}$$

$$x \in Z_+^n, \tag{4}$$

where $J(x)$ is the output signal of the circuit under consideration, Lb and Ub are the bounds (lower and upper) of $J(x)$, x_j, $j = 1,\ldots,n$; are the parameters of the elements, which values have to be optimized, and which accept discrete values only. Z_+^n is the set of nonnegative integral n-dimensional vectors, l_j and u_j are bounds of x_j, such that $(u_j–l_j)/(2u_j).100 = tol_j$ determines the tolerance of x_j. Here m is a positive integer number equal to the number performed Monte Carlo simulations by means of statistical simulator IESD (see [3]) on the circuit under consideration. The optimal solution of **(1-4)** is F=0.

The optimal tolerancing problem can be defined as:

$$\textbf{Max} \quad F_T = (\min_{j=1,\ldots,n} tol_j) \tag{5}$$

$$\textbf{subject to:} \quad \sum_{i=1}^{m} \delta_i = 0 \tag{6}$$

$$\delta_i = \begin{cases} 1 \text{ if } J(x) \notin [Lb, Ub]; \\ 0 \text{ if } J(x) \in [Lb, Ub]; \end{cases} \tag{7}$$

$$u_j = x_j + x_j \,.\, tol_j / 100 \tag{8}$$

$$l_j = x_j - x_j \,.\, tol_j / 100 \tag{9}$$

$$l_j \le x_j \le u_j; \quad j = 1,\ldots,n; \tag{10}$$

$$tol_j \in T; \quad T = \{1, 2, 5, 10, 15\} \tag{11}$$

$$x \in Z^n, \tag{12}$$

where the variables tol_j are the tolerances of parameters x_j, $j=1,\ldots,n;$ and x_j, $j=1,\ldots,n;$ Lb, Ub, are beforehand known constant integers.

II. THE DESIGN CENTERING METHOD IN A REDUCED SEARCH SPACE DCSDR

The problem (1)-(4) is a combinatorial one (see constraints (4)) and belongs to the class of NP-hard optimization problems. For this reason the statistical optimization methods solving this problem could be very time consuming. Usually the complete enumeration of the permitted range options is large. A heuristic has been proposed in order to limit the trials number (see [2]). At each step best and worst performance randomly generated circuits are used to define the values and the tolerances at the next step. This approach is effective, but also could be rather inefficient when the number of circuit elements n is large and great number of parameter variations must be investigated. To improve the efficiency a direct search method for fast design centering (FDC) was proposed in [4]. It is based on the Nelder and Mead's method [6], known as the most efficient among the direct search methods. FDC method is modified correspondingly for the discrete search space. The drawback of FDC method is that the number of simplex vertices is (n+1), that is greater than the number of circuit elements n, and for large n great number of Monte Carlo simulations should be performed.

[1]Galia I. Marinova is with the Technical University of Sofia, Faculty of Telecommunications, Sofia -1000, Bul. "Kliment Ohridski", N°8, Bulgaria, E-mail: gim@tu-sofia.bg ;

[2]Vassil G. Guliashki is with the Institute of Information Technologies, Sofia-1113, "Acad. G. Bonchev" Str., Bl. 29A, Bulgaria, E-mail: vgoul@iinf.bas.bg ;

Performing design centering it turns out that very often the circuit output $J(x)$ depends stronger on the variation of some "sensitive" circuit elements than on the variation of the left over elements. This feature of the problem (1)-(4) makes it possible to use reduced search space, optimizing only the "sensitive" elements. Based on this idea a design centering method with space dimension reduction (DCSDR) solving the problem (1)-(4) was proposed in [5]. In this way the search process is enhanced considerably. The search step in DCSDR method is calculated like that one in FDC method. To keep the constraints (4) the step components are rounded off to integer values. In contrast to the FDC method, which uses simplex with (n+1) vertices, where n is the number of circuit elements, the simplex in DCSDR method has (k+1) vertices, where k is the number of "sensitive" circuit elements. Another way used to make the search procedure in DCSDR faster than that one in FDC is to perform Monte Carlo analysis not in each simplex vertex like the FDC method, but only at the initial step and at the end of each DCSDR optimization iteration.

Other specific feature of DCSDR method is that the initial simplex is constructed not as trivial regular simplex (see [6]), but in different way. Let $J(x)$ be the output signal function of the circuit and Jd be the desired value of the output signal. To evaluate the steps quality during the DCSDR iteration the measure of deviation from the desired value is used: $D(x) = |J(x)-Jd|$. Let among the vector variations obtained after the initial Monte Carlo analysis x^B be the vector, for which $D(x^B)$ has minimal value and let x^W be the vector, for which $D(x^W)$ is maximal. The vectors x^{Mi}, $i = 1, \ldots, n$; are constructed, where $x^{Mi}_j = \{ x^B_j$ if $i \neq j$; x^W_j if $i = j\}$. As starting simplex vertices are chosen vector x^B and k vectors among x^{Mi}, $i = 1, \ldots, k$; for which the deviation of $J(x)$ is relative great, i. e. the k vectors corresponding to variations of "sensitive" elements. It is expected, that the optimization by reflection the worst simplex vertex through the weight center of the left over simplex vertices with objective - minimizing $D(x)$ will move the simplex to a search space region, where its weight center would improve the F value.

The steps of the method are presented as follows:

Step 0. Give the starting value of:

- iterations limit *itlim*, where *itlim* > 0,
- the vector of elements parameters $x^{(0)}$,
- l_j and u_j for each parameter x_j , $j = 1, \ldots, n$;
  The tolerance value $tol_j = 100.(u_j - l_j)/(2.u_j)$;
- the lower and the upper bound for the output signal *Lb* and *Ub*. The desired output signal value is usually $Jd=(Lb+Ub)/2$.

Set the iteration counter *icount* = 0.

Step 1. Perform a Monte Carlo analysis in $x^{(icount)}=x^{(0)}$. Let $D(x)$ has obtained its minimal and maximal value correspondingly in vectors x^B and x^W during the analysis. In case $F(x^{(icount)})=0$, go to *Step* 13. Otherwise go to *Step* 2.

Step 2. Use x^B and x^W to construct the vectors x^{Mi}: $x^{Mi}_j = \{ x^B_j$ if $i \neq j$; x^W_j if $i = j\}$, $i = 1, \ldots, n$.

Step 3. Calculate the rate of influence on the output signal value for the change of each x_i, $i = 1, \ldots, n$; : $\sigma_i = |[(J(x^{Mi}) - J(x^B))/(J(x^W) - J(x^B)]/[(x^W_i - x^B_i)/x^B_i]|$. Create an index set I from indices i, for which $\sigma_i > \theta$, where $\theta \in (0, 1)$ and could be different for different circuits. Usually $\theta = 0{,}03$. Let there are k such indices in I, where $k \leq n$.

Iteration:

Step 4. Fix the values of all elements, which indices $i \notin I$. Construct a nonregular simplex in the reduced k-dimensional space, determined by the left over elements. The simplex has k+1 vertices. One of them is x^B and the others are x^{Mi}, $i \in I$.

Step 5. Find among the k+1 simplex vertices the vector $x^{(j)}$, for which $D(x^{(j)}) = \max D(x^{(i)})$, $i = 1, ..., k+1$;

Step 6. Find the weight center of the k left over simplex vertices:

$$\boldsymbol{x_c} = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}^{(i)} \qquad (13)$$

Step 7. Construct the vector:

$$\boldsymbol{x = x^{(j)} + \lambda(x_c - x^{(j)}),} \qquad (14)$$

where $\lambda = 2$; (In case the last found vector x at *Step* 7 is again that one with max $D(x)$ start *Step* 7 with $\lambda = 1.5$;) Round off each component x_j to the closest feasible discrete value (see (**4**)), so that the constructed vector $x \in Z^n_+$. In case $D(x) < D(x^{(j)}$, replace $x^{(j)}$ by x and go to *Step* 5, otherwise change λ (simplex contraction): $\lambda=1.5$ and construct a new vector x using (**14**). Round off each x_j to the closest feasible discrete value. In case $D(x) < D(x^{(j)}$, replace $x^{(j)}$ by x and go to *Step* 5, otherwise change λ (simplex contraction): $\lambda=1.25$ and construct a new vector x using (**14**). Round off each x_j to the closest feasible discrete value. In case $D(x) < D(x^{(j)}$, replace $x^{(j)}$ by x. Go to *Step* 8.

Step 8. Find the weight center of all simplex vertices:

$$\boldsymbol{x_t} = \frac{1}{k+1}\sum_{i=1}^{k+1} \boldsymbol{x}^{(i)} \qquad (15)$$

Step 9. Set *icount*=*icount*+1. Set $x^{(icount)} = x_t$. Perform a Monte Carlo simulation in $x^{(icount)}$. In case $F(x^{(icount)})=0$, go to *Step* 13.

End of the Iteration

Step 10. In case the last iteration was unsuccessful, i. e. $F(x_t)$ has greater value than the last found $F(x)$, and if there are components $i \in I$, for which x_i has not being reflected towards the corresponding component x^B_i during the iteration, then construct a vector $p = \Sigma(x^{(icount-1)} - x^{Mi})$. Let among the simplex vertices $x^{(j)}$ be the vector, for which $D(x^{(j)}) = \max D(x^{(i)})$, $i=1,..., k+1$;

Construct the vector:

$$\boldsymbol{x = x^{(j)} + p} \qquad (16)$$

and perform a Monte Carlo analysis in it.

Step 11. If *icount* = *itlim* go to *Step* 13.

Step 12. Let $D(x)$ has obtained its minimal and maximal value correspondingly in vectors x^B and x^W during the simulation. Construct the vectors x^{Mi}: $x^{Mi}_j = \{x^B_j$ if $i \neq j$; x^W_j if $i = j\}$, $i = 1, \ldots, n$. Go to *Step* 4.

Step 13. End of the search process.

To complete the statistical optimization technology, the design centering should be followed by an optimal tolerancing search procedure.

III. Illustrative Examples

A. Design centering of voltage regulator circuit

The voltage regulator circuit from [4, 5] is used to illustrate the performance of DCSDR method (see Fig. 1). The parameters that can be optimized are the resistors R_1, R_2, …, R_6 and the capacitor C_1. Two applications of the voltage regulator circuit are studied: one for 24 V stabilized output voltage and another one for 15 V stabilized output voltage.

- The output voltage of the circuit should be constant 24 ± 0.5 V and the desired value *Jd* = 24.00 V. This case was studied in [4, 5]. The objective here is to minimize *F* (see (**1**)) until *F*=0. The Monte Carlo analysis is performed with 100 randomly generated circuits with given tolerances.

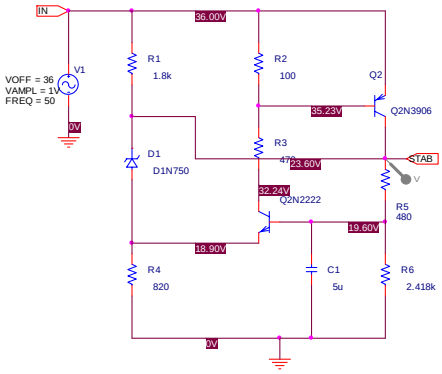

Fig. 1. Voltage regulator circuit

After a Monte Carlo analysis with 1% tolerance for all R, C elements the circuit output $J(x)$ did not violate the *Lb* and *Ub* bounds in all the 100 cases. When 2% tolerance of elements values was chosen, 3 from 100 randomly generated circuits had an output voltage above 24.5 V, i. e. *F*=3, when *m*=100, tol_j = 2%, j = 1,…,7;. The results obtained after one DCSDR iteration are as follows: R_1 = 1810Ω, R_2=100Ω, R_3=472Ω, R_4=820Ω, R_5=476Ω, R_6=2451Ω and C_1=5μF. The optimal tolerances found for the corresponding circuit elements are: 15%, 15%, 15%, 10%, 2%, 2%, 15%.

- The circuit from Fig. 1 was tested here subject to another output voltage constraints: The output voltage should be constant 15 ± 0.15 V and the desired value *Jd* = 15.00 V. In this case the constraints are stronger than in the first one. The tolerances are accepted to be 1% for all circuit elements. At the initial solution 2 from 100 randomly generated circuits had an output voltage above 15.15 V, i. e. *F*=2, when *m*=100, tol_j = 1%, j = 1,…,7;. The elements which have influence on the circuit output value are R_2, R_4, R_5 and R_6. At *Step* 3 are calculated: σ_2 = 1.68, σ_4 = 3.04, σ_5 = 39.76, σ_6 = 39.23. As sensitive are accepted the elements R_5 and R_6. The reduced dimension of the search space is 2. The results obtained after the first DCSDR iteration are presented in Table 1.

Higher tolerance values for the found solution x_t lead to fail.

TABLE 1

Solution	*F*	R1 [Ω]	R2 [Ω]	R3 [Ω]	R4 [Ω]	R5 [Ω]	R6 [Ω]	C1 [μF]
$x^{(0)}$	2	1800	100	470	820	449	1356	5
x^B		1800	100	470	820	455	1355	5
x^W		1798	100	472	824	452	1363	5
$x^{(1)}\lambda$=2		1800	100	470	820	461	1372	5
$x^{(2)}\lambda$=1.5		1800	100	470	820	460	1359	5
x_t	0	1800	100	470	820	459	1362	5

B. Design centering of pulse controlled voltage regulator

A pulse controlled voltage regulator from [1] is used as second experimental circuit. It is presented on Fig. 2

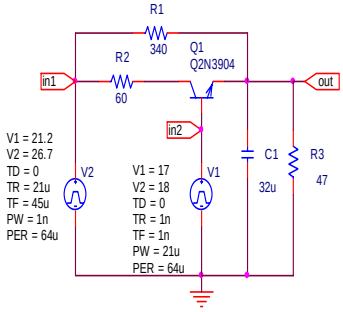

Fig. 2. Pulse controlled voltage regulator

The parameters that can be optimized are the resistors R_1, R_2, R_3 and the capacitor C_1. The output voltage of the circuit should be constant 15 ± 0.15 V and the desired value *Jd* = 15.00 V. The tolerances are accepted to be 10% for all circuit elements. At the initial solution 53 from 100 randomly generated circuits violated the constraint on the output voltage value, so *F* = 53. At *Step* 3 are calculated: σ_1 = 0.57, σ_2 = 1.81, σ_3 = 3.7, σ_4 = 0.027. As sensitive are accepted the elements R_2 and R_3. The reduced dimension of the search space is 2. The first DCSDR iteration was unsuccessful. The method from [5] would stop the optimization without finding a better solution. A look at the output voltage histogram from Fig. 3 (upper one) could explain this drawback. The histogram of the stabilized output voltage value for the initial design is close to Gaussian distribution and means, which is 14.987 V, is very close to the desired value of 15 V. To overcome the mentioned drawback the new *Step* 10 is added to DCSDR method. The design centering continues following the improved method. The fail is reduced for the optimal circuit values from 53% to 33%. The histogram for the stabilized output voltage for the optimized circuit is presented on Fig. 3 (the lower one). The means in this case is 15.027 V.

The distribution in the optimized design is similar to the initial one, but slowly shifted to the right and thus follows better the constraints. The optimized element values will be a better start point for a future optimal tolerancing and will permit 100% yield (0% fail - F=0) with lower tolerance reduction, than in the initial case. The results for the optimization steps are presented in Table 2.

TABLE 2

Solution	F	R1 [Ω]	R2 [Ω]	R3 [Ω]	C1 [μF]
$x^{(0)}$	53	340	60	47	32
x^{B}		341	60	47	32
x^{W}		340	57	37	34
$x^{(1)}$ λ=2		340	57	57	32
$x^{(2)}$ λ=1.5		340	59	45	32
$x^{(3)}$ λ=2		340	58	49	32
x_t	59	340	58	48	32
x	33	340	61	49	32

The histograms of the circuit before and after design centering from Fig. 2 are presented on Fig. 3.

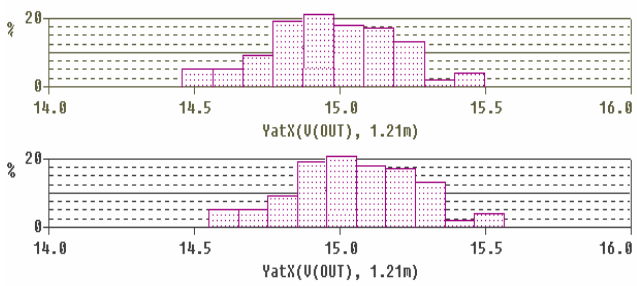

Fig. 3. Histograms for the output voltage of pulse controlled voltage regulator before and after design centering

IV. Conclusion

The paper presents the improved version of DCSDR method for design centering using search space with reduced dimension. Comparing the efficiency of FDC method [4] and of DCSDR method [5], the last method is more efficient because it performs (*itlim*+1) Monte Carlo analyses and *itlim*.(*k*+1) evaluations of $J(x)$. The FDC method performs [(*n*+1) + *itlim*] Monte Carlo analyses. For the test experiments here one Monte Carlo analysis generates 100 random circuits, i. e. 100 evaluations of $J(x)$ are calculated. Usually $n > k$ and the iteration limit *itlim* is a small positive integer. The improved DCSDR method performs one more Monte Carlo analysis if Step 10 is used at the end of calculations, but this method is able to solve design centering tasks when the initial design values are close to the optimal ones. This case was unapproachable by the method from [5]. The new method was tested on three different circuit designs, which confirmed its advantages compared to the former version.

Acknowledgement

This research is partly supported by The National Science Fund, MON, Bulgaria, Project I-1203/2002

References

[1] G. I. Marinova, D. I. Dimitrov "Statistical estimation of multiple realizations in radiocommunication modules synthesis", proposed for ICEST'2005, June 29 - July 1, Nis, Serbia and Montenegro, 2005.

[2] G. I. Marinova, D. I. Dimitrov "Statistical analysis and optimization of voltage regulator circuit using IESD and ORCAD environment", ICEST'2003, October 16-18, Sofia, Bulgaria, 2003, pp. 478-482.

[3] G. Marinova "Statistical design simulator for electronic design – IESD", *The European Design and Test Conference*, *Users Forum*, Paris, March 11-14, 1996, pp. 297.

[4] G. Marinova, V. Guliashki, "Fast design centering method for electronic circuits", Proceedings of 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services TELSIKS'2003, Nis, Serbia and Montenegro, 2003, pp. 701-704.

[5] G. Marinova, V. Guliashki, D. Dimitrov, "A design centering method with space dimension reduction", Proceedings of ELEKTRONIKA'2004 Conference, Sofia, Bulgaria, 21-22. 05. 2004, pp. 134-139.

[6] G. V. Reklaitis, Ravindran A., Ragsdell K. M. „*Engineering Optimization. Methods and Applications*", John Wiley and Sons, 1983.

[7] Zhang J. C. and M. A. Styblinski, "*Yield and variability optimization of integrated circuits*", Klewer Academic Publishers, 1995.