



**INAOE**

# **Variational Analysis of Analog Integrated Circuits based on Symbolic Analysis**

By

**Adolfo Adair Palma Rodríguez**

A Dissertation submitted in partial fulfillment of the requirements for the degree of:

**MASTER OF SCIENCE WITH MAJOR ON ELECTRONICS**

at the

**Instituto Nacional de Astrofísica Óptica y Electrónica**

August 2012

Tonantzintla, Puebla

Advisors:

**Dr. Esteban Tlelo Cuautle**

INAOE

**Prof. Sheldon X.-D. Tan**

University of California Riverside

©INAOE 2012

The author hereby grants to INAOE permission to reproduce and to distribute copies of this Thesis document in whole or in part.



*Al único Dios que es el Camino, la Verdad y la Vida*

*A mis padres Adolfo y Abigail que siempre me han amado.*

*A mis hermanitas, Meli y Mariel.*

*Al amor de mi vida, Fabiola.*



# Acknowledgements

First, I want to thank God for his free, sovereign, unconditional, electing love, and providence. I can now say: Thus far the Lord has helped me. I want to give thanks also to my family and Fabiola for their prayers, unconditional love and support.

I want to express my gratitude to my advisor, Dr. Esteban Tlelo Cuatle, for his support during the process of this Thesis and for his guidance in the important process of research publications. Also I want to give thanks to my co-advisor Prof. Sheldon X.-D. Tan for giving me the opportunity to work in his research group in MSLAB at University of California Riverside

I would like to express my gratitude also, to the National Institute of Astrophysics, Optics and Electronics (INAOE) to give me the opportunity to have studied into an institution of academic excellence.

Finally, I want to acknowledge to the Consejo Nacional de Ciencia y Tecnologia (CONACyT), for the financial support during my master studies. This work is partially supported by UC-MEXUS and CONACYT under grants CN-11-575 and 131839-Y.

# Abstract

It is well known that with the down-scaling of the integrated circuit (IC) technology, nanometer circuit designs become more and more sensitive to process variations. It is caused in part by matching requirements between transistors. Those variations are produced by fluctuations at the moment of manufacturing and have been continuously increasing in relative magnitude as IC technology continues to scale to 45nm and below. That is why the integrated circuit designer must study these process variations in order to obtain more and more reliable circuits. Analog circuit designers usually perform a Monte-Carlo (MC) simulation to analyze the stochastic mismatch and to predict the variational responses of their designs under faults. As MC analysis requires a large number of repeated circuit simulations, its computational cost is very expensive. In this Thesis, a new performance bound analysis of analog circuits considering process variations is proposed. The new method applies a recently reported graph based analysis approach to generate the symbolic transfer function of a linear(ized) analog circuit. As a result of applying symbolic analysis, one can have a good understanding on the behaviour

of the analog IC with respect to its circuit elements, and for that same reason, repeated simulations as in MC analysis aren't needed, because the analytical expression is helpful in determining performance bounds. For instance, in this Thesis, the frequency response bounds (maximum and minimum) are obtained by performing nonlinear constrained optimization, where the magnitude of the transfer function is the objective function to be optimized subject to some ranges of process variational parameters. As shown in this Thesis, the response bounds given by the optimization method are very accurate. Furthermore, experimental results on testing several analog ICs show that the proposed method gives the correct bounds, as those provided by the Monte Carlo analysis, while the time process is comparable to the Monte Carlo simulation.

# Resumen

Es bien sabido que con el escalamiento de la tecnología de circuitos integrados (IC), los circuitos de dimensiones nanométricas se vuelven más sensibles a las variaciones de proceso, debidas en parte a los requerimientos de acople entre transistores. Estas variaciones son producidas por fluctuaciones al momento de fabricarlos y se han incrementado debido al continuo escalamiento de la tecnología por debajo de los 45nm. Es por eso que los diseñadores de circuitos integrados deben estudiar estas variaciones de proceso para obtener circuitos cada vez mas confiables. Los diseñadores analógicos usualmente realizan el análisis Monte-Carlo (MC) para analizar la disparidad estocástica y para predecir las respuestas variacionales de sus diseños bajo fallas. Dado que el análisis MC requiere un gran número de simulaciones repetidas, su costo computacional es alto. En esta Tesis, un nuevo método de análisis variacional para circuitos analógicos considerando variaciones de proceso es propuesto. Este nuevo método aplica un análisis basado en grafos, reportado recientemente, para generar la función de transferencia simbólica de un circuito analógico linealizado. Como resultado de aplicar análisis simbólico, se puede

tener un buen entendimiento del funcionamiento de un IC analógico con respecto a sus elementos de circuito, y por esa misma razón, no se requiere hacer simulaciones repetidas como en el análisis MC, porque la expresión analítica es útil para obtener los bordes de comportamiento. Por lo pronto, en esta Tesis, los límites de la respuesta en frecuencia (máximo y mínimo) son obtenidos por medio de optimización con restricciones, en las que la magnitud de la función de transferencia es la función objetivo a ser optimizada sujeta a algunos rangos de los parámetros dados por las variaciones de proceso. Como se muestra en esta Tesis, los límites obtenidos de este método son muy exactos. Además, resultados experimentales para varios IC analógicos muestran que el método propuesto calcula los límites correctos, como los obtenidos en el análisis MC, mientras que el tiempo de ejecución es comparable a la simulación MC.



# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Resumen</b>	<b>vi</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Variational Analysis . . . . .	3
1.2 Symbolic Analysis . . . . .	6
1.3 Objectives of the Thesis . . . . .	7
<b>2 Symbolic Formulation by Directed Graph</b>	<b>9</b>
2.1 Simple Case: Without Node Reuse . . . . .	11
2.2 Advance Case: Node Reuse . . . . .	13
2.2.1 The Advanced Case in detail . . . . .	15
2.3 Symbol Factorization and Symbolic Derivative . . . . .	18

	1
2.3.1 Factorization of Symbol $W$ . . . . .	18
2.3.2 Symbolic Derivative with respect to $W$ . . . . .	19
2.3.3 Symbolic Mixed Derivative . . . . .	20
<b>3 Symbolic Analysis with MOSFET based analog circuits</b>	<b>21</b>
3.1 Implementation of the Graph-Based Approach . . . . .	21
3.1.1 Parsing the netlist . . . . .	22
3.1.2 Small Signal Models and NullOr Equivalents . . . . .	26
3.1.3 Matrix equations formulation . . . . .	34
3.1.4 Experiments . . . . .	36
<b>4 Constrained Optimization</b>	<b>41</b>
4.1 A simple example in MATLAB . . . . .	44
4.2 Line Search . . . . .	48
4.3 Projected Gradient Method . . . . .	49
4.4 Spectral Projected Gradient Method . . . . .	51
4.5 Remarks . . . . .	53
<b>5 Experimental Results</b>	<b>54</b>
<b>6 Conclusions</b>	<b>60</b>

# Chapter 1

## Introduction

It is well known that due to both complicated chemical and physical processes at the moment of manufacturing a circuit in today's microelectronic devices, differences between the designed circuit and the manufactured product are present [1]. Among these differences in manufacturing one can find variations in critical dimensions and inter-level dielectric thicknesses which lead us to mismatch. Variations have a significant impact in analog circuits since these become more and more sensitive to process variations given the number of matching requirements between transistors. Manufacturing variations, as mismatch, are random in nature, and an increasing effort is aimed in order to minimize these effects. Nevertheless, fluctuations at the moment of manufacturing cannot be eliminated and have been continuously increasing in relative magnitude as integrated circuit (IC) technology continues to scale to 45nm and below, owing to the increasing

process-induced variability [2, 3]. According to [4, 5] mismatch in CMOS devices nearly doubles for each process generation less than 90nm due to an inverse-square-root law dependence with the transistor area.

## 1.1 Variational Analysis

Nowadays, designers must deal with variations to guarantee the correct functioning of the IC design after its manufacture. On the one hand, traditional corner based verification can not meet the required accuracy. On the other hand, Monte-Carlo (MC) is based on semi-classical physics, and it is well known for its MC based statistical simulation capabilities that have become the standard for statistical analysis and yield estimation in design of digital and analog integrated circuits under process variations, due to its advantage of high accuracy and generality [6]. It provides approximated solutions to a variety of mathematical problems by performing statistical sampling experiments and both can be applied to problems with absolutely no probabilistic content as well as to those with inherent probabilistic structure. In MC method, values are assigned to the circuit components, based on a population distribution of each component. Then the values of a number of performance parameters are calculated and compared with standard performance parameter acceptability limits [7]. However, simple Monte-Carlo method is expensive and slow and its inefficiency may lead to the bottleneck of analog circuit optimization. For this reason, many fast Monte-Carlo methods have been proposed

in order to improve the efficiency of classical Monte-Carlo methods. Among these approaches Importance Sampling Monte Carlo (ISMC) [8, 9, 10] can be found where one uses an alternative sampling distribution that makes the interesting samples more likely than under the original distribution. The twist in the distribution is then corrected for by weighting the samples with the so called likelihood ratio [11]. Other approach is Latin Hypercube Sampling Monte Carlo (LHSMC) [12], where LHS is a mathematical technique pioneered by McKay [13] restricted to monotonic functions only, and later generalized to n-dimensions by Keramat [14] and applicable to any function. This LHSMC method is similar to the simple MC except in samples generation step where LHS is used. Also, it is very simple and does not involve any further simulations and has smaller variance with respect to simple MC method [14]. One last fast MC approach is Quasi Monte Carlo (QMC) [15, 16] based method which improves the convergence rate compared with simple MC method. QMC uses a different class of sampling methods called low-discrepancy sequences (LDS), or also called Halton sequences which are deterministic with no random component where the points in the sequence are generated to satisfy some rigorous notion of uniform coverage of the sampling space [6].

Some efforts have been made using non Monte Carlo methods for statistical analysis [17]. Performance bound methods emerged as attractive techniques for statistical analysis. Bounding or worse case analysis of analog circuits under parameter variations has been studied before for fault-driven testing and tolerance analysis of analog circuits [18, 19, 20]. Some frequency domain performance bound methods were proposed in

[21, 22] to compute the lower and upper bounds of transfer functions' of magnitudes and phases. Method in [21] applies Kharitonov's functions to obtain the performance bounds in frequency domain, but no systematic method was proposed to obtain variational transfer functions. This method has been improved by [22] where symbolic analysis approach was applied to derive exact transfer functions and affine interval method was used to compute variations transfer functions. However, the affine interval method can lead to over conservative results.

In this Thesis, a non Monte-Carlo based method is presented based on performance bound analysis technique in frequency domain. This new method first drives the exact transfer functions of linear (linearized) analog circuits via a graph-based symbolic analysis method , which is described in Chapters 2 and 3. Then the frequency response bounds of the transfer functions in terms of magnitude and phase are obtained by a constrained nonlinear optimization technique described in Chapter 4. Constrained optimization problems involve a function  $f(x)$  which is minimized or maximized and called the objective function, in this case, the transfer function's magnitude of a given circuit is the objective function and the constraints are the elements of the circuit, i.e. resistances, capacitances or inductances, and they can be as many as the user needs depending on the number of elements in the circuit.

## 1.2 Symbolic Analysis

Symbolic analysis is a formal technique to calculate the behavior or a characteristic of a circuit with its variables (dependent and independent) and the circuit elements represented by symbols [24]. This technique had a growing interest between 1960's and 1980's because of the increasing computing power and that many computer analysis techniques were proposed. A growing interest from the circuit design community started in late 1980's and this interest is seen in the success of modern symbolic analyzers as one can see in programs as ISAAC [25, 26] , ASAP [27, 28] , SYNAP [29, 30], SAPEC [31], SSPICE [32], SCYMBAL [33] , SCAPP [34], and GASPAP [35].

Knowledge of the behavior of a circuit is very important in the design process. For this reason, symbolic simulators become to be a very useful tool as they give as the result, the analytic expression in a closed form of the circuit representing the relationships between its parameters, this being an advantage over numerical simulators. Also symbolic analysis is useful when many numerical cancellations lead us to large roundoff error. However, the behavior of a circuit can be verified accurately in a numerical simulation, this is specified only for a set of parameter values contrary to the symbolic simulation where the returned expression is valid even if the parameter values change (as long as the circuit topology remains the same).

Symbolic Analysis can be categorized in different approaches: Tree Enumeration methods, Signal Flow Graph Methods and Determinant based methods [36]. Tree

Enumeration methods were proposed by Maxwell and Kirchhoff and first used for analysis of RCL networks represented as weighted undirected graphs. The advantage of this method is that the expressions are irreducible, however, for RLC- $g_m$  networks the tree enumeration cannot be applied directly and tradeoffs between sign calculation and obtaining cancellation-free product terms are encountered [37]. Unlike Tree Enumeration methods, Signal Flow Graphs are weighted directed graphs but the product terms obtained by this method are still not irreducible or cancellation-free. Determinant Based Methods are also not cancellation-free but it has been proven that topological methods (such as Tree Enumeration methods and Signal Flow Graph methods) don't offer any advantage over this methods. Any circuit transfer function can be obtained by applying Cramer's rule to the set of linear equations but solving the determinants in order to obtain the transfer function makes this an exponential space complexity method, in spite of that, this drawback can be mitigated by applying recursivity.

### 1.3 Objectives of the Thesis

In this Thesis a new method to deal with variational analysis will be shown based on constrained optimization and an evaluation of the feasibility of this approach to analog integrated circuits is also given. A graph-based symbolic tool will be explained and used in order to obtain the symbolic transfer function. Constrained optimization methods used in this work are based on line search methods [38, 39, 40, 40, 41].



Chapter 2 talks about symbolic formulation of the AC transfer function of analog circuits based on directed graphs and how one can, when the graph is already built from the Nodal Admittance (NA) Matrix, factorize any expression and from that one can directly obtain the symbolic derivative and symbolic mixed derivative . In Chapter 3 it will be explained how NullOr elements i.e., Nullator and Norator, are introduced in the graph-based symbolic tool in order to represent MOSFET transistors and non-NA compatible elements, a detailed explanation on how the NA matrix is built is also explained as the matrix equations formulation of any circuit. Some experiment results are given in this last chapter. In Chapter 4 a brief introduction on Constrained Optimization will be given, and the different methods used in this Thesis will be explained. The process of the proposed algorithm is shown in this last Chapter and will be explained step by step with a simple MATLAB example on an RLC circuit using an active set method. In Chapter 5 plots of the lower and upper bounds obtained by applying the method proposed approach to several examples are shown and discussed. Finally, conclusions of this Thesis will be given in Chapter 6.

## Chapter 2

# Symbolic Formulation by Directed Graph

In order that the whole discussion in this Thesis being self-contained, the following text in this chapter has been taken from [42]. A very useful and studied tool for the formulation of symbolic expressions by solving a system of equations via Cramer's rule is the so called Determinant Decision Diagram (DDD). Thus Determinant Decision Diagram (DDD) based methods are classified as determinant based methods. DDDs are adapted from a data structure called Zero Suppressed Binary Decision Diagram (ZBDD) created in order to manage sparse subset systems. Two important observations on which determinant decision diagrams are based are that the admittance matrix representing a circuit is sparse and that a symbolic expression often shares many subexpressions

[43]. In order to formulate the DDD, the determinant has to be expressed as a sum of products (SOP) which then are classified via a ZBDD in which every sum is a subset. Thus, DDD is an indirect method as it involves extra processing and the formulation of the full determinant beforehand.

In this chapter an approach for the solution of a system of equations based on simple graph manipulations is introduced and compared to the well known DDD method. For instance, the graph structure in use is directly formulated from the matrix with no intermediate steps. Important observations on which this graph representation was developed are that the admittance matrix (in this case the nodal admittance matrix described in [44]) representing a circuit is sparse and that a symbolic expression often shares many subexpressions, characteristics that the DDD representation shares in principle. The determinant representation by applying the proposed graph-based technique is compact, unique and the complexity to obtain the symbolic expression depends on the size of the graph which in turn depends on the non-zero entries of the admittance matrix.

The code for all the functions was implemented in C and compiled in a RedHat Linux environment with the GNU C compiler gcc-4.2, Quad-Core Intel with 24GB RAM. ANSI-C compliance was considered of importance for migration purposes and universality, non-compliant functions were avoided.

## 2.1 Simple Case: Symbolic Determinant without node reuse

The circuit size is a challenge in performing symbolic analysis because a large number of symbolic terms are manipulated. Fortunately, this problem is mitigated when applying a graph-based approach. In such a case, every determinant has a unique representation, and is liable to symbolic manipulations. To understand how this approach works, let's consider the following determinant [37]:

$$\det(M) = \begin{vmatrix} a & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & h \\ 0 & 0 & i & j \end{vmatrix} = adgj - adhi - aefj - bcgj + cbih \quad (2.1)$$

If the determinant's size is  $n \times n$ , we expect to have paths of  $n + 1$  levels. So if there is a path that is not complete, i.e. that does not have  $n + 1$  levels or that has a zero in any element of the path, it will be eliminated completely since this means that this expression is multiplied by zero. This structure is built in a depth-first fashion. Every element of the graph structure corresponds to a non-zero entry in the admittance matrix. As a result, for this particular example one obtains the graph shown in Fig. 2.1, where all multiplications by zero were already omitted.

Path eliminations are performed by sending a prune signal if a zero is found inside the path. This prune signal is propagated all the way up until a summing point is

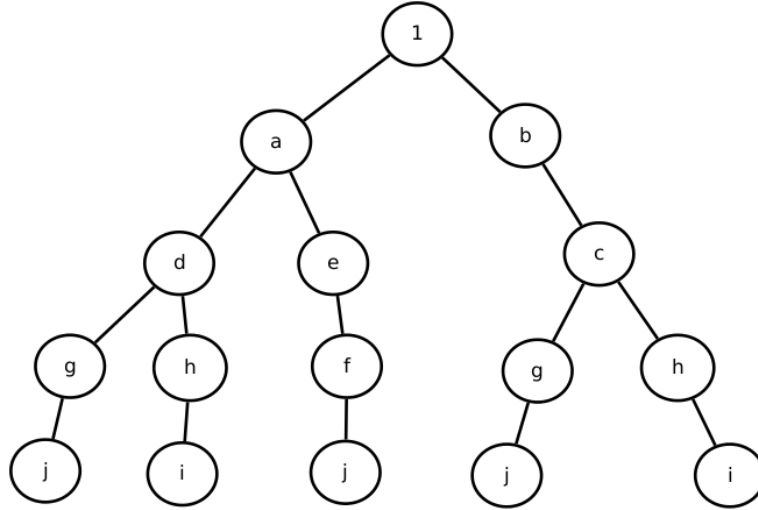


Figure 2.1: Graph representation of Equation (2.1)

reached so the whole branch does not form part of the final structure and the path is terminated to zero instead. As one sees, this graph structure implemented is a tree in which the arithmetic operations are encoded in the depth of the tree node, that is, different depth implies multiplication while equal depth implies addition. This leads us to get the expression:

$$\det(M) = a[d(gj - hi) + e(-fj)] + b[c(-gj + hi)] \quad (2.2)$$

A key point is related to the assignation of signs to each node in the expanded graph. They are established by applying the rule of signs from Cramer's rule. When applying graph methods to evaluate a determinant, not only one can obtain a factorized exact

symbolic expression, but also derive all transfer relationships with respect to each node, and in a post processing step to each branch circuit variable. The graph representation shown on this section is compact for a large class of analog circuits.

## 2.2 Advanced Case: Symbolic Determinant with node reuse

If the determinant is expressed as a SOP in graph form with no reuse of information the result is a graph with many repetitive terms (Fig. 2.1) corresponding to minors of the matrix that are repeated as was shown in the previous section. The smallest the matrix minor, the highest the repetition rate (if they are not zeroed out previously). In Fig. 2.1 the terms  $(g \cdot j)$  and  $(h \cdot i)$  are repeated twice. The information in each node is exactly the same, the only difference being the sign which in the end is trivial as it can be easily computed.

The main idea in the algorithm presented in this Thesis is that it is possible to reuse the information of those repeated nodes. For the previous example it is possible to consider the products  $(g \cdot j)$  and  $(h \cdot i)$ , and for correctness sake also the product  $(f \cdot j)$  (which are the three non-zeroed product terms for the determinant of the  $2 \times 2$  minors of  $M$  in equation (2.1)) as independent subgraphs. So now we have five two-nodded subgraphs with vertex sets  $V_1 = \{g, j\}$ ,  $V_2 = \{h, i\}$ ,  $V_3 = \{f, j\}$ ,  $V_4 = \{g, j\}$  and  $V_5 = \{h, i\}$ . An important observation is that even though it may seem that edges  $E_1 = \{g, j\}$  and  $E_4 = \{g, j\}$  are equivalent as they convey the same operation which is

the product  $(g \cdot j)$ , so that the subgraphs  $G_1 = \{V_1, E_1\}$  and  $G_4 = \{V_4, E_4\}$  carry the same information and can be considered equivalent.

The reusing was possible because the information which made the graphs  $G_1$  and  $G_4$  different is somehow stripped from the subgraphs and obtained from somewhere else. It is possible to extend this idea even more in order to reuse the nodes in the last row by identifying the information which make nodes with same terms different from each other. The difference between the subgraphs  $G_1 = G_4$  and  $G_3 = \{V_3, E_3\}$  is the ancestor of node  $j$ . That is, for  $G_1 = G_4$  it's node  $g$  so the subgraph represents the operation  $g \cdot j$  whereas subgraph  $G_3$  represents  $f \cdot j$ . Remember that each node is linked to a non-zero matrix entry<sup>1</sup> so there is a node for every  $A_{\text{row}, \text{col}} \neq 0$ , node  $f$  is in turn a representation of  $A_{3,2}$  and node  $g$  is  $A_{3,3}$ . When visiting node  $j$  the question is "*Was the previous node  $f$  or  $g$ ?*". When selecting a pivot  $A_{\text{row}, \text{col}}$  while formulating the determinant, row and column are removed, reformulating the previous question into "*Have we already visited a node from the same column as  $f$ ?*" if the answer is affirmative then the previous node was  $g$ .

Extending this same procedure to the whole graph, the first obvious consequence is that there are no repeated nodes, in other words, for a matrix  $A$  with  $nz$  non-zero entries, there are  $nz$  nodes. So far information conveyed by relationships in the graph from Fig. 2.1 concerning to the sign of the adjugate matrix and the already visited columns has been stripped. The adjugate matrix sign can be computed as given by

---

<sup>1</sup>It's sometimes useful to break a node representing  $A_{\text{row}, \text{col}}$  term into two or more parallel nodes.

Equation (2.3) requiring to know row and column. Row is given already as the depth of a node making necessary to store the information pertaining to the column into the node.

$$\text{sign} = (-1)^{\text{row}+\text{col}} \quad (2.3)$$

The symbolic manipulations performed are agnostic of the origin of the matrix, that is, the only condition is that the matrix be square. Each node in the graph structure corresponds to a matrix non-zero entry, which in turn encapsulates the summation of one or more circuit elements.

### 2.2.1 The Advanced Case in detail

Three different data structures are required. The first and most obvious is the node structure which contains the following fields:

- Node Name: A unique name for the given node. It is assigned as an index number.
- Terms: An array containing the index and sign of the element (mapping it to the element table described in section 3.1.1, in page 22).
- Column: The column of the non-zero entry in the matrix that this node belongs to.
- Descendants: An array of node pointers linking to the descendants of the current node in the graph structure.



The second data structure is a graph type. It is useful to have a graph structure with the fields:

- Graph name: A unique name for the current graph. It is possible to have many different graphs. At least two graphs are required at first: numerator and denominator.
- Matrix size: The size of the matrix. Only one dimension is required as the input is expected to be a square matrix.
- Root node: The root of the graph. In the simple AC analysis the root node is a trivial node with term value equal to 1 and column and row equal to zero. When multiple graphs are constructed (i.e. Factorization) the root node can be any of the  $nz$  nodes.
- Visited Columns: When traversing a graph to formulate the determinant the column of a visited node is appended to this array .

The third and final structure is a matrix type. This structure stores not only the input matrix but also the independent vector.

First of all,  $nz + 1$  nodes are created ( $nz = \text{non-zero entries}$ ) and their respective structure fields filled in with the information read directly from the matrix. The Row and Column fields are useful to compute the sign rule when formulating the determinant as well as to determine which nodes are to be skipped.

The graph is built starting with a trivial node named 0 with term value of 1. Remember the multiplication is codified as depth in the graph. The different nodes are linked accordingly. The algorithm to build the graph structure for the representation of  $|A|$  is shown as Algorithm 1. With the graph already built, the expression for

---

**Algorithm 1** buildGraph( $A(i, j)$ ,  $Ancestor$ )

---

```

 $m \leftarrow$  number of Columns of  $A$ 
 $n \leftarrow$  number of Rows of  $A$ 
 $D \leftarrow$  set of descendants of  $Ancestor$ 
Ensure:  $m > 0 \wedge n = m$ 
function BUILDGRAPH( $A$ ,  $Ancestor$ )
  if  $m > 1$  then
     $prune = 1$ 
    for  $i = 1$  to  $m$  do
      for all Columns  $j$  that  $A_{i,j} \neq 0$  do
        if BUILDGRAPH( $C_{ij}$ ,  $A_{i,j}$ )  $\neq 0$  then
           $D = D \cup A_{i,j}$ 
           $prune = 0$ 
    else
      if  $A_{i,j} \neq 0$  then
         $prune = 0$ 
  return  $prune$ 

```

---

the determinant is then formulated as in Algorithm 2. The graph structure for the determinant of the matrix as seen in (2.1) is shown in Fig. 2.2.

---

**Algorithm 2** Symbolic Determinant from graph

---

1. Read the graph in a modified DFS fashion:
    - (a) Keep track of which columns have been visited
    - (b) Skip nodes from columns already visited
  2. Symbolic expression is the product of a node times the summation of its visited descendants
-

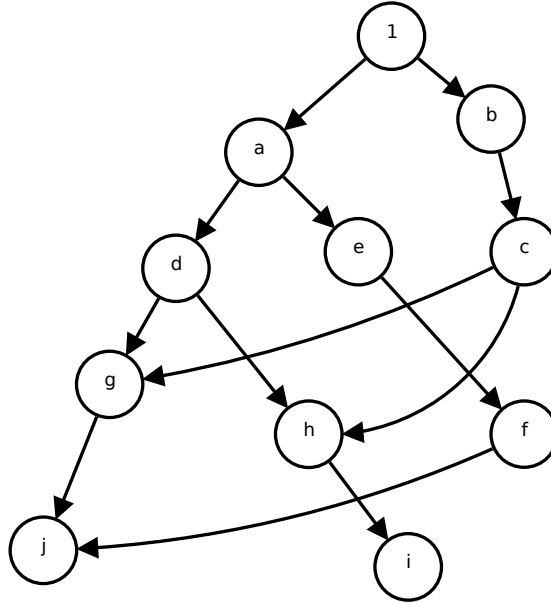


Figure 2.2: Determinant graph

## 2.3 Symbol Factorization and Symbolic Derivative

### 2.3.1 Factorization of Symbol $W$

Symbol factorization is useful in order to facilitate numerical evaluation of the expression and in order to obtain the symbolic derivative with respect to a given symbol. Factorization takes place by following a simple algorithm (Algorithm 3). The result is the expression of the determinant as a polynomial represented by an array of sum of products with one entry for each power of symbol  $W$  with non-zero coefficient.

Suppose one have a matrix (2.4) and we want the polynomial form as powers of  $w$ . The resulting graph is shown in Fig. 2.3 and the polynomial expression is in (2.5).

---

**Algorithm 3** Graph to Polynomial

---

1. Expand the nodes of the graph containing the symbol  $W$
  2. Read the graph in DFS fashion and preserve only those routes from root to bottom with at least one occurrence of symbol  $W$ .
  3. The number of occurrences of the symbol is the power of  $W$  for a given route.
  4. Each route is then expressed as a sum of products replacing the symbol for a one.
  5. The summation of all appended routes for each power of  $W$  forms the coefficients.
- 

$$M = \begin{bmatrix} a+w & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & h \\ 0 & 0 & i & j \end{bmatrix} \quad (2.4)$$

$$w^0 = adgj - adhi - aefj - bcgj + bchi$$

$$w^1 = adi + dgj - dhi - efj - bci \quad (2.5)$$

$$w^2 = di$$

### 2.3.2 Symbolic Derivative with respect to $W$

The derivative  $\frac{\partial |\mathbf{A}|}{\partial W}$  is straightforward as symbol  $W$  in the expression  $|\mathbf{A}|$  has already been factorized. The graph expansion can be performed in the original structure generated when obtaining the determinant thus reducing the memory footprint. Expansions for more than one symbol are possible.

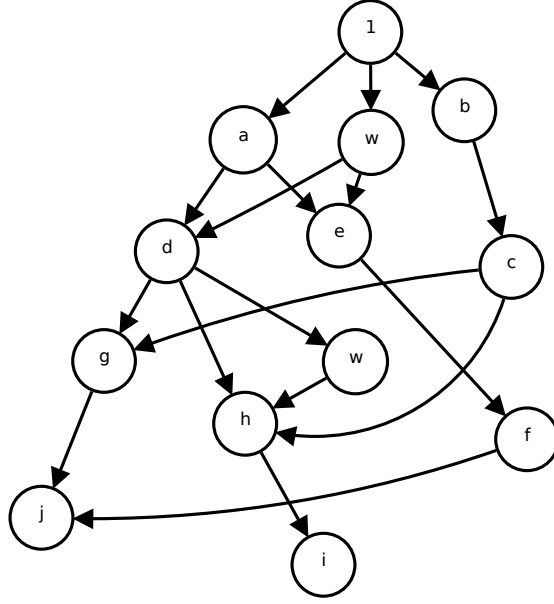


Figure 2.3: Expanded graph

### 2.3.3 Symbolic Mixed Derivative

It is possible to get a mixed derivative of the form  $\frac{\partial^2 f}{\partial W_1 \partial W_2}$  for symbols  $W_1$  and  $W_2$ . For this purpose we make use of the resulting symbolic expression obtained previously in Section 2.3.2. Such expression is in the form of a sum of products, which is further separated in  $n$  coefficients (powers of  $W_1$  with coefficient 0 omitted) of  $W_1$ . Each coefficient is expressed as a graph, thus obtaining  $n$  graphs. The derivative procedure given in Section 2.3.2 is repeated for all  $n$  graphs individually now for the symbol  $W_2$ . This is useful for some optimization problems where the Hessian matrix needs to be computed [45].

## Chapter 3

# Symbolic Analysis with MOSFET based analog circuits

### 3.1 Implementation of the Graph-Based Approach

The proposed graph-based approach for the solution of a system of equations starts off with a SPICE netlist as input. The elements implemented are R, C, L, V, I, E, G and M, being resistor, capacitor, inductor, independent voltage source, independent current source, voltage controlled voltage source, voltage controlled current source and MOSFET. Both independent voltage and current sources can be DC, AC or both.

The netlist is to be used as a way to input the circuit topology along with circuit elements and values. If the numerical evaluation of the resulting symbolic transfer functions generated is required, the small signal values for the parasitic elements and the

operating point conditions are read from the output listing in HSPICE<sup>TM</sup> format. This numeric values are of use when evaluating the transfer function to verify correctness.

The flow of the tool is shown in Fig. 3.1

The derivative is computed in an alternative module which is bypassed when the only output required is the actual transfer function. The algorithm to implement this symbolic derivative accommodates for consecutive derivatives with respect to different variables (symbols). The flow of this module is presented in Fig. 3.2.

### 3.1.1 Parsing the netlist

The first step is to parse the netlist and build suitable data structures for each group of elements. The symbol given to every device from this point onwards is the same as its name. To keep consistency, the symbol name is taken exactly as specified in the netlist (ex. R\_name, C\_name, M\_name, etc) input is not case sensitive and is parsed as lowercase. Elements are first grouped into one of four different tables: conductances, independent sources, controlled sources and Mosfet Table 3.1.

Values for elements and sources are considered strings of characters so the tool is agnostic of whether they are actually numbers or functions until the numerical evaluation of the output is performed. The symbol name is always the element identifier and it's name exactly as it appears in the netlist. The network relationships of the different elements are codified in their node connections. Because it is a lot easier to treat nodes as integer indexes, a mapping is created first by building an array with all the nodes as

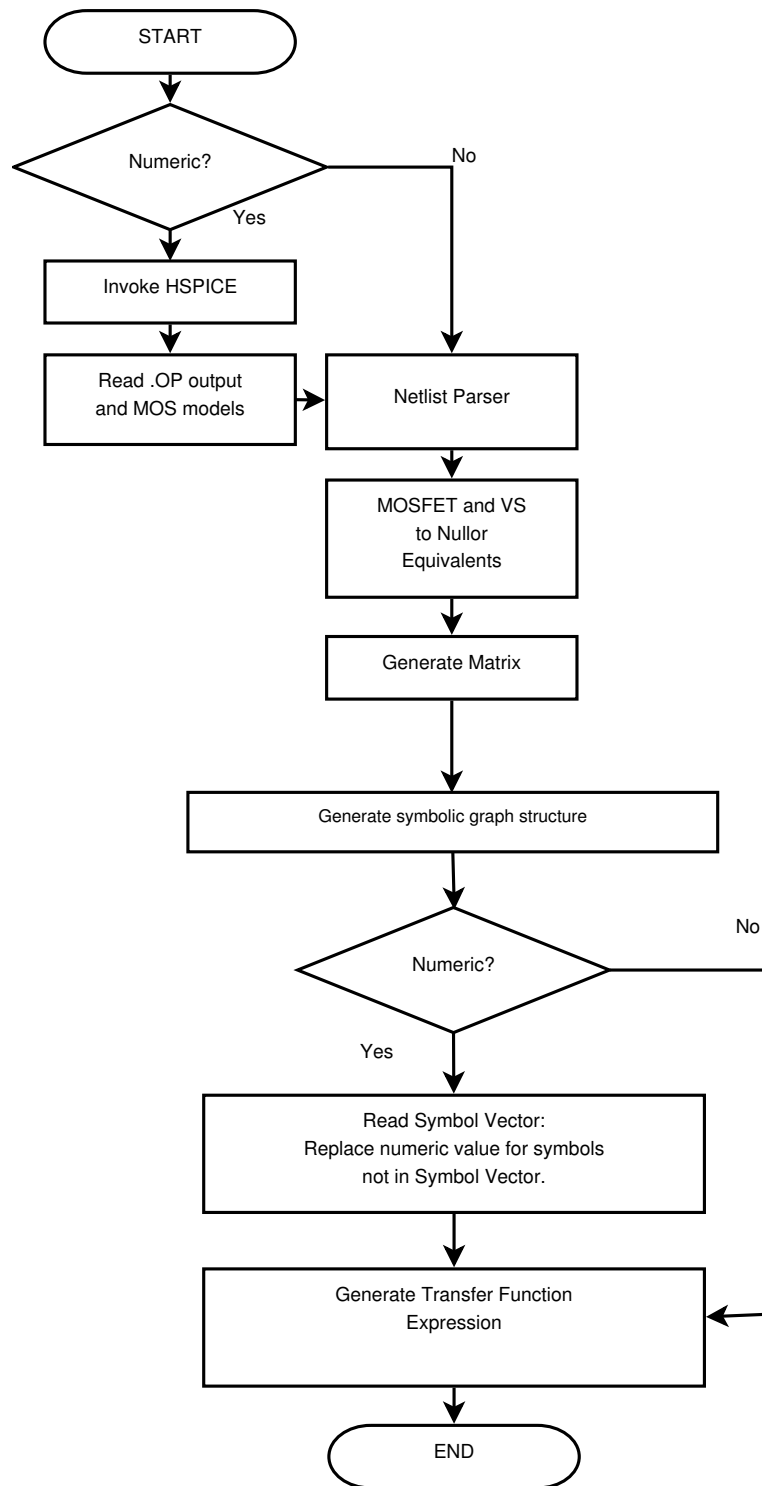


Figure 3.1: Symbolic Tool Flow



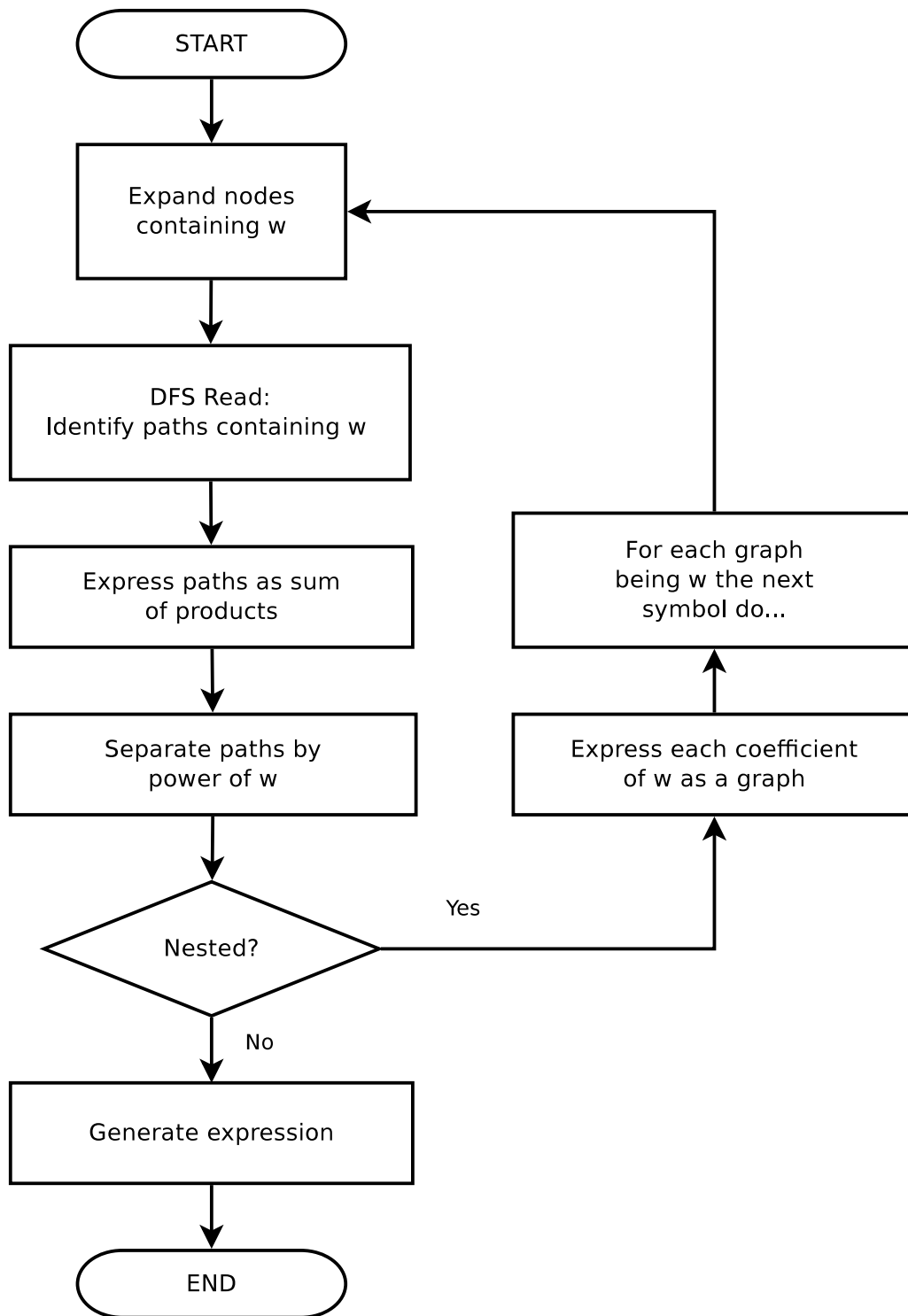


Figure 3.2: Module for Symbolic Derivative

Table 3.1: Elements Tables

Table Type	Fields
Conductances	Name, Node 1, Node 2, Value
Independent Sources	Name, Node 1, Node 2, DC, AC
Controlled Sources	Name, Node 1, Node 2, Node 3, Node 4, Gain
MOSFET	Name, Drain, Gate, Source, Bulk, Width, Length, ModelName

they appear in the netlist and any duplicate occurrences are removed.

The tool is intended to perform AC analysis so it is of no use to keep independent sources with only DC value, as they are only valid in a DC or transient context. When removing voltage DC sources their nodes are collapsed, while current DC sources nodes are left floating. Collapsing two nodes in a circuit is equivalent to assigning to both nodes the same mapping and so is done in this step. If the numerical evaluation of the resulting transfer function is to be performed, the output listing (.lis file for HSPICE™) is read and the values for the operating point are parsed and stored in an array with the same index correspondence as the element symbol array so that the mapping between the symbols and their numerical values is direct.

Up to this point the elements are separated in groups according to their type, nodes are collected in a single array and are mapped to indexes to facilitate their treatment,

DC sources are removed.

### 3.1.2 Small Signal Models and NullOr Equivalents

In the present implementation, active elements are substituted by their controlled source based small signal model. In turn, controlled sources are modeled with combinations of norator and nullator in order to make use of the extensive studies of NullOr based circuits. The Nullator and Norator are abstract elements with ideal characteristics [36]. The voltage across the nullator terminals is zero and does not allow current to flow through it. Even though it has some properties of an open circuit it doesn't have an immittance or a scattering representation.

$$V_1 = V_2 = \text{arbitrary}, I_x = 0 \quad (3.1)$$

For the norator, the voltage between its terminals is arbitrary and an arbitrary current can flow through it. The norator also shows some properties of an open circuit. The circuit representations of the nullator and norator are shown in Fig. 3.3.

$$V_1 \neq V_2 = \text{arbitrary}, I_x = \text{arbitrary} \quad (3.2)$$

One can combine these elements in order to obtain a new one called Nullor where, when it is modeled as a two-port element, its first port is the Nullator and the second is the Norator, that is why the Norator and Nullator receive the name of Nullor elements. The Nullator was first introduced by Carlin [46]. This simple element has proven its usefulness in areas like symbolic analysis. One of the main advantages of the Nullor

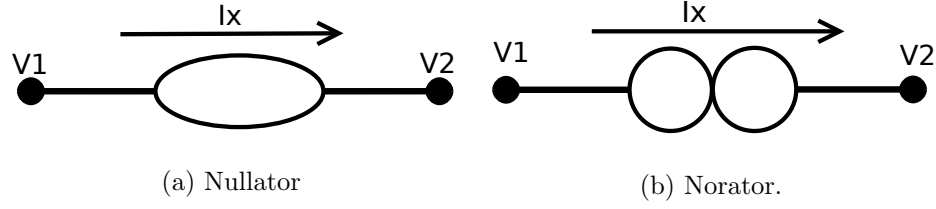


Figure 3.3: Pathological element symbol for a) Nullator and b) Norator.

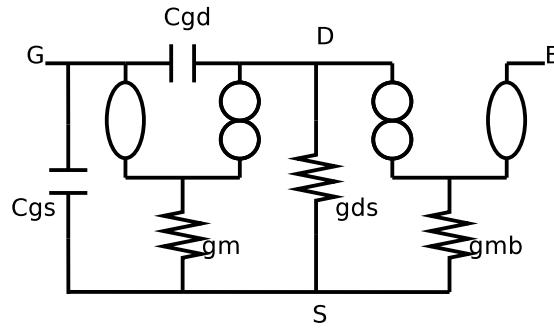


Figure 3.4: Nullor representation of the MOSFET

elements is that nodal analysis (NA) of an active circuit is simple and also they can model active circuits independently of the particular realization of the active devices [36, 47]. For the MOSFET, the Nullor equivalent including the parasitic gate-source and gate-drain parasitic capacitances used in the sensitivity analysis is shown in Fig. 3.4, therefore for an operational transconductance amplifier (OTA) its Nullor equivalent would be as shown in Fig. 5.1

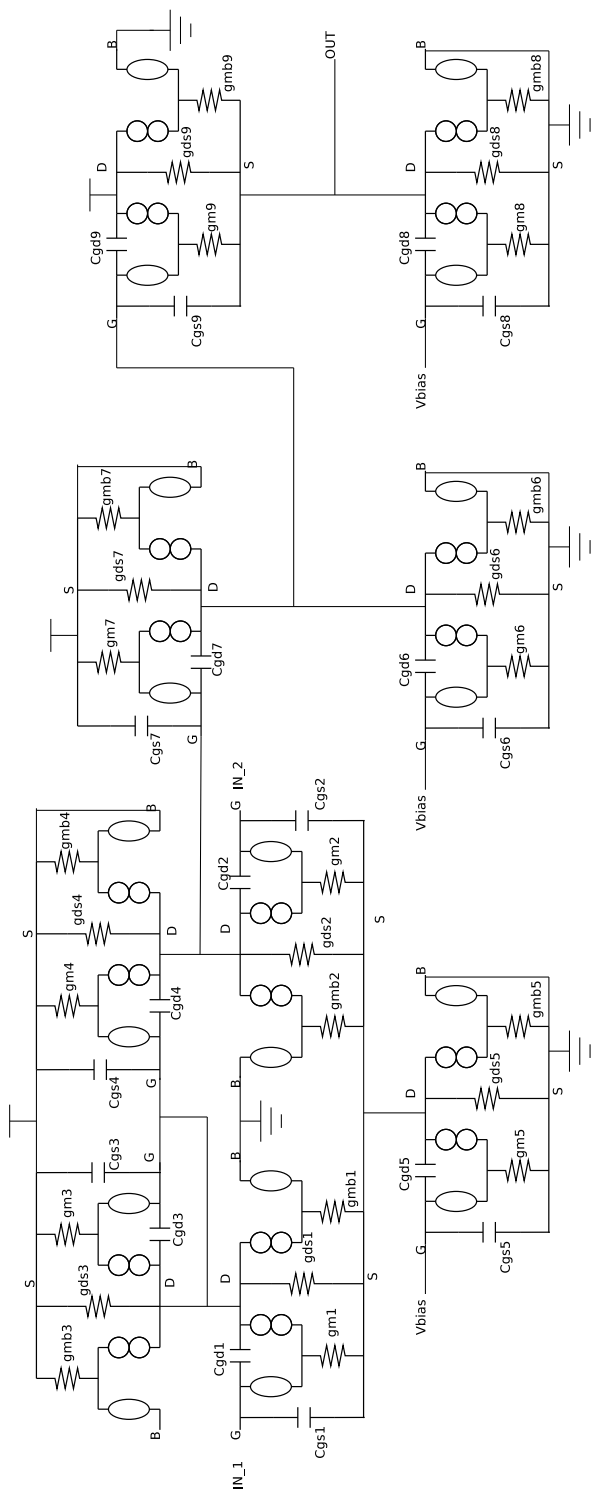


Figure 3.5: Nullor representation of a three stage OTA

The guidelines for obtaining the nodal admittance matrix by applying nodal analysis is summarized below for convenience and is explained in detail in [44, 48].

Two sets of pairs of nodes are formed, one for ROWs and one for COLs. These two sets are then used to form the admittance matrix by performing the Cartesian product of every subset. In the ROW group a subset is formed for every node with no Norator connected to it, and a different subset for every group of nodes connected by floating Norators. In the COL group a subset is formed for every node with no Nullator connected to it, and a different subset for every group of nodes connected by floating Nullators. Two groups of admittances are formed, the first (group A) containing a subset for every node listing all the admittances connected to it, the other (group B) listing floating admittances with the corresponding pair of nodes. If a node is present in a subset in ROWs and a subset in COLs then the corresponding subset of admittances (from group A) is summed at the matrix position (ROW index, COL index). If a pair of nodes is present, one in a subset of ROWs and the other in a subset of COLs, the corresponding admittance (from group B) is summed with negative sign at the matrix position (ROW index, COL index).

Since the method applied is symbolic NA, the independent voltage source is transformed to a current source equivalent circuit. In order to simulate a differential input in circuits like the differential pair and the three stages operational transconductance amplifier (OTA) a voltage controlled voltage source which is converted also to a nullor equivalent is used. The nullor representation of both elements traditionally non-NA

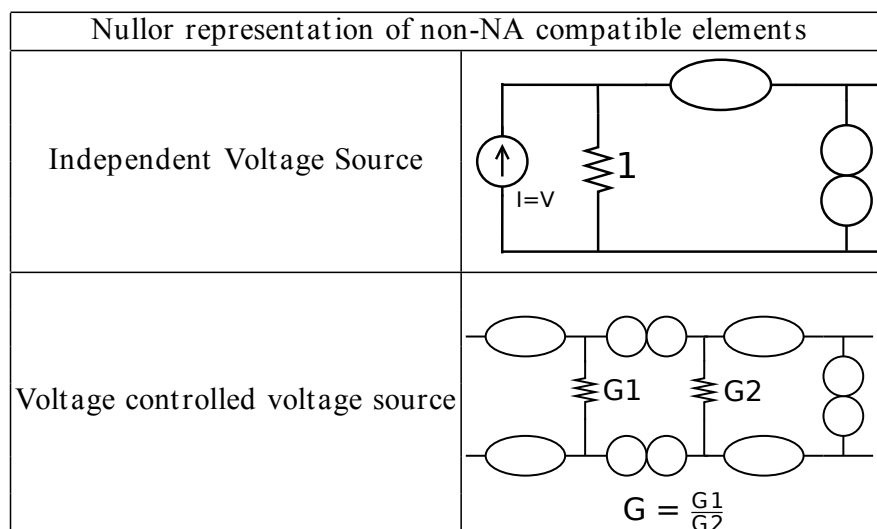


Figure 3.6: Nullor representation of non-NA compatible elements

compatibles is shown in Fig. 3.6.

In a previous section it was explained why symbolic modeling is intended to give an insight into certain behaviors and tendencies of the circuit in question. With this in mind, it is now evident that the more complex the small signal model of a device the more accurate is the resulting simulation but at the cost of increasing machine operation time. A compromise can be reached where the qualitative behavior is roughly preserved at the expense of numerical accuracy. Implemented are three levels of parasitic elements for the small signal model for the Mosfet:

- Level 0 has no parasitic elements and models only the voltage controlled current source with gate-source as the controlling branch voltage and transconductance  $gm$ .

- Level 1 accounts for level 0 plus  $C_{gs}$ ,  $C_{gd}$  and  $G_{ds}$ .
- Level 2 accounts for level 1 plus the voltage controlled current source whose controlling branch voltage is bulk-source with transconductance  $g_{mb}$ .

From this first stage the elements are classified as *NA compatible* or *NA incompatible* according to their small signal models. Incompatible elements are then treated as their respective Nullator/Norator (Nullor) equivalents as shown before in Fig. 3.6.

There are two reasons why elements can be incompatible. Firstly if the dependent variable of the function for a given element is voltage, as is the case for voltage sources and secondly if the independent variable for the element function is current. The reason behind this, as the reader may already be aware, is that in nodal analysis the unknown vector is composed of voltages (node voltages) while the independent vector is formed by current sources. Then, the need arises to apply a manipulation such that those conditions are preserved while maintaining a physical equivalence.

On the other hand, a fully differential OTA can be modeled as a single VCCS, which in turn can be viewed as a pair of MOSFET transistors in differential input configuration Fig. 3.7.

The ideal conditions for each of the two MOSFETs are kept: infinite input impedance (zero branch current in input Nullator), zero output impedance (arbitrary voltage) and ideal gm. Parasitic elements can then be arranged around this basic block.

As explained before a set of Nullator-Norator equivalents are implemented to account



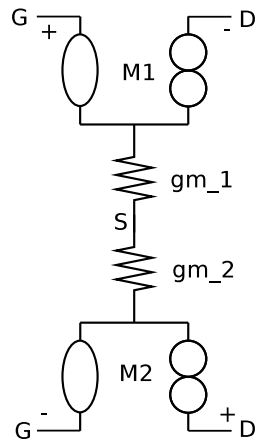


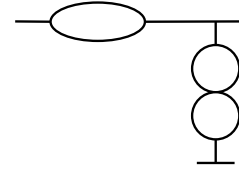
Figure 3.7: Nullor Equivalent of the MOSFET Differential Pair

for the intrinsic voltage controlled current source of the Mosfet small signal model Fig. 3.4, voltage sources and controlled sources Fig. 3.6.

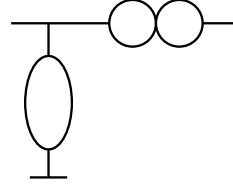
The basic building blocks for the equivalents are the Voltage Follower 3.8a and the Current Follower 3.8b. When replacing a certain element with its NullOr equivalent there are three basic operations that have to be performed:

1. Add an element to the proper array (nullator, norator, conductance or independent source)
2. Add extra nodes if required (MOSFET, Voltage Source, etc.)
3. Assign unique name to each element and the extra nodes (as required)

Adding a new element is as easy as appending the new entry to the corresponding structure (conductance, independent source, nullator or norator) and updating the



(a) Voltage Follower



(b) Current Follower

Figure 3.8: Nullor based (a) Voltage Follower and (b) Current Follower

number of elements for the given structure. As a list of the nodes originally read from the netlist is kept, it is easier to keep track of the nodes added afterwards if any new node is appended at the end of the structure containing the numerical mapping. When adding an element it is useful if the name is a composition of the name of the original element it is bound to. For instance, if *CGS* is included into the model for the MOSFET named  $M_1$ , the name of the new parasitic capacitor can be *cgs\_1*. In this way it becomes easier to tell to which netlist element a given symbol belongs to.

Now that there is a mapping of unique node elements, it is possible to begin replacing elements for their corresponding NullOr equivalents. For the case of the MOSFET the small signal model conformed with the selected parasitic elements and the corresponding NullOr is replaced in the circuit netlist.

### 3.1.3 Matrix equations formulation

Now that we know the node mappings we can create an adjacency matrix for each of the elements. From this adjacency matrix the NA formulation is performed by following the method in [44] which is reproduced here in a short form for convenience (Algorithm 4 [42]).

---

**Algorithm 4** NA Formulation with NullOr elements

---

1. Two groups of nodes are formulated, one named ROW and another named COL. The admittance matrix is formulated by performing the Cartesian product of these two groups.
  2. Formulate ROW group of nodes:
    - Add a subset for each node with no Norator connected to it.
    - Add a subset for every chain of nodes connected by floating Norators.
  3. Formulate COL group of nodes:
    - Add a subset for each node with no Nullator connected to it.
    - Add a subset for every chain of nodes connected by floating Nullators.
  4. Group A: Admittances seen in a node.
    - A set for each node containing all admittances connected to it.
  5. Group B: Floating Admittances.
    - A set for each floating admittance containing its pair of nodes.
  6. Formulate Matrix
    - If a node is present in a subset in ROWs and a subset in COLs then the corresponding subset of admittances (from group A) is summed at the matrix position (ROW index, COL index).
    - If a pair of nodes is present, one in a subset of ROWs and the other in a subset of COLs, the corresponding admittance (from group B) is summed with negative sign at the matrix position (ROW index, COL index).
-

This procedure is shown for a simple common source circuit as the one shown in Fig.

3.9.

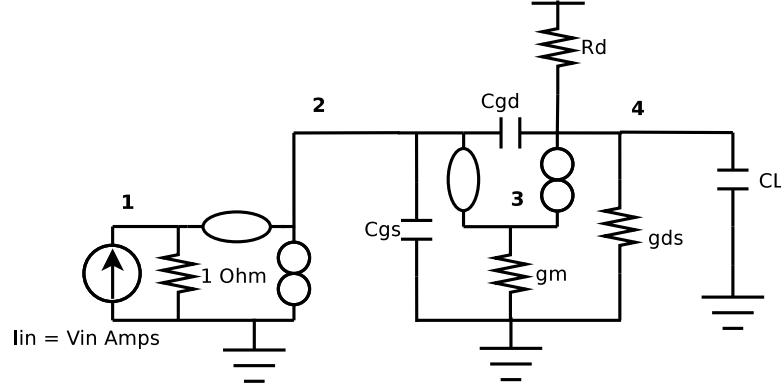


Figure 3.9: Common Source Small Signal Equivalent

Table 3.2: NA Formulation

ROWS	COLS	FLOATING	SEEN BY NODE
(1)	(1,2,3)	(2,4) : $sC_{gd}$	1 : $1Ohm$
(3,4)	(4)		2 : $sC_{gs}, sC_{gd}$ 3 : $gm$ 4 : $sC_{gd}, g_{ds}, \frac{1}{R_d}, sC_L$

The formulated groups are shown in Table 3.2. The resulting matrix is a 2 x 2 system as shown in (3.3).

$$\begin{array}{cc}
 & \begin{matrix} (1, 2, 3) & (4) \end{matrix} \\
 \begin{matrix} (1) \\ (3, 4) \end{matrix} & \begin{bmatrix} 1 & 0 \\ gm - sC_{gd} & -(sC_{gd} + g_{ds} + \frac{1}{R_d} + sC_L) \end{bmatrix} \begin{bmatrix} v_{(1,2,3)} \\ v_{(4)} \end{bmatrix} = \begin{bmatrix} I_{in} \\ 0 \end{bmatrix}
 \end{array} \quad (3.3)$$

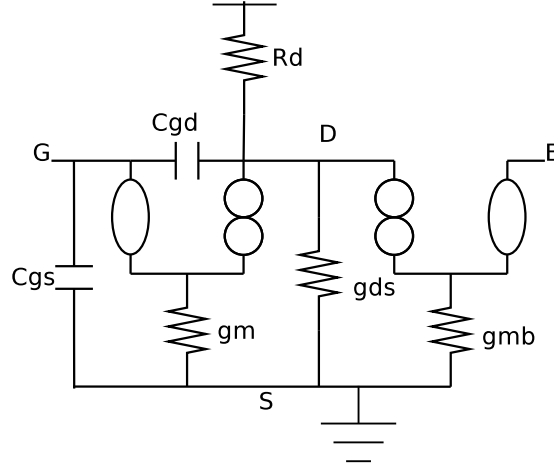
The solution for this system of equations becomes an application for the symbolic tool presented in this Thesis. The system of equations is solved by using Cramer's rule by computing  $n + 1$  determinants for  $[v_1, v_2, \dots, v_n]^T$  node voltages in order to completely define the circuit. For the system of equations  $\mathbf{A}x = b$  where  $\mathbf{A}$  is  $n \times n$  matrix, the solution by Cramer's Rule is given as  $x_i = \frac{|\mathbf{b} \rightarrow \mathbf{A}_i|}{|\mathbf{A}|}$ .

### 3.1.4 Experiments

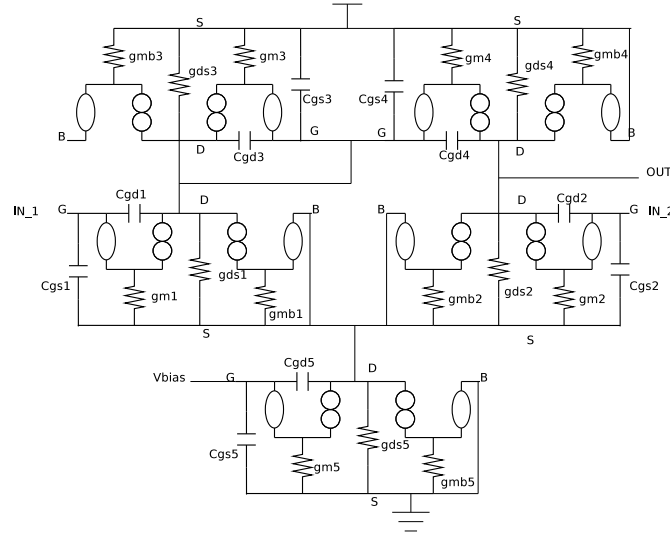
For the experimental verification the symbolic transfer function of four CMOS and one BJT (small signal model with Rpi, Cpi, Cmu and Gm VCCS) circuits was computed, and time metrics were taken for matrix equations formulation, denominator, numerator and transfer function computation. The test cases are the differential pair and common source shown in Fig. 3.10, the three stages uncompensated OTA in Fig. 5.1 [49], recycling folded cascode OTA [50] and 741 OPAMP respectively as shown in Table 3.3.

The test circuit for the UA741 is a Small-Signal model with R, C and VCCS elements. This same circuit was tested with SCAD3 [37] which does not support reading MOST from netlist. The running and evaluation time reported by SCAD3 is of 2.97s while the proposed symbolic tool takes 2.7043s.

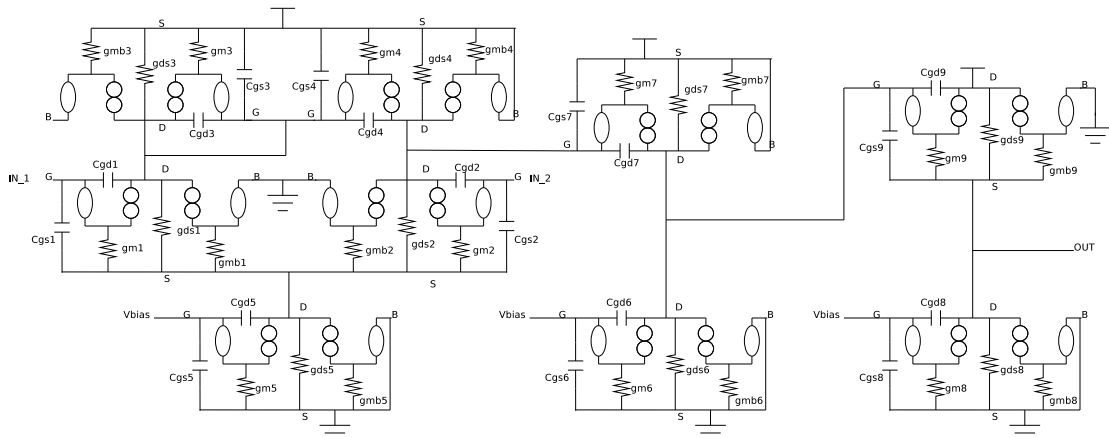
Numerical evaluation of the resulting symbolic expression is performed in order to provide a comparison between a well known and mature technology (HSPICE<sup>TM</sup>), a previously developed symbolic tool [43] and the proposed graph-based symbolic tool. The test circuit is the same Differential Pair from Fig. 3.10b where the input to SCAD3



(a) Common Source.



(b) Differential Pair.



(c) Three stages uncompensated amplifier.

Figure 3.10: Nullor equivalentents of the amplifier circuits.

Table 3.3: Symbolic Formulation and Numerical Evaluation of  $D(s)$ ,  $N(s)$  and  $H(s)$ 

Circuit Features			Computer Time (seg)			
Circuit	Elements	Nodes	Equations	$D(s)$	$N(s)$	$H(s)$
Differential Pair	35	26	1.1235	0.122	0.1464	1.4895
RFC OTA [49]	106	56	1.6603	0.201	0.1869	2.2633
LV Amp	33	18	2.35	0.058	0.0464	2.4544
Common Source	8	6	0.8581	0.041	0.0205	0.9811
741	112	77	0.5123	1.37	0.822	2.7043

has been converted to VCCS equivalents. The AC analysis output is shown in Fig. 3.11.

Numerical evaluation of the resulting expressions for the RFC OTA [49] are shown in Fig. 3.12.

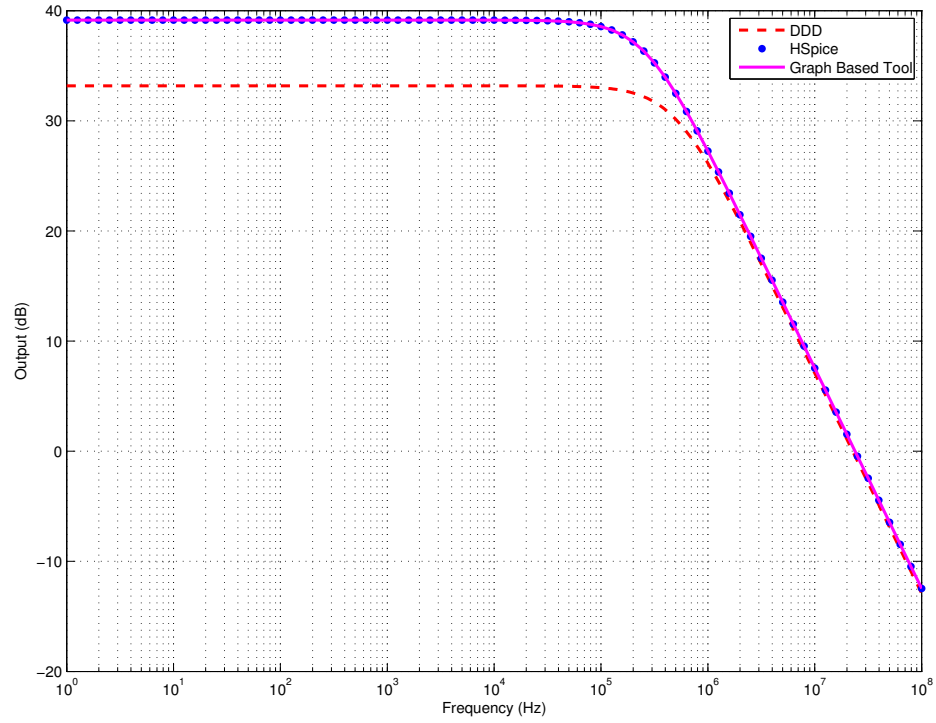


Figure 3.11:  $H(s)$  comparison between DDD (SCAD3) [43] (dashed), HSPICE<sup>TM</sup> (dots) and Graph Based Tool (solid) for the differential pair topology.



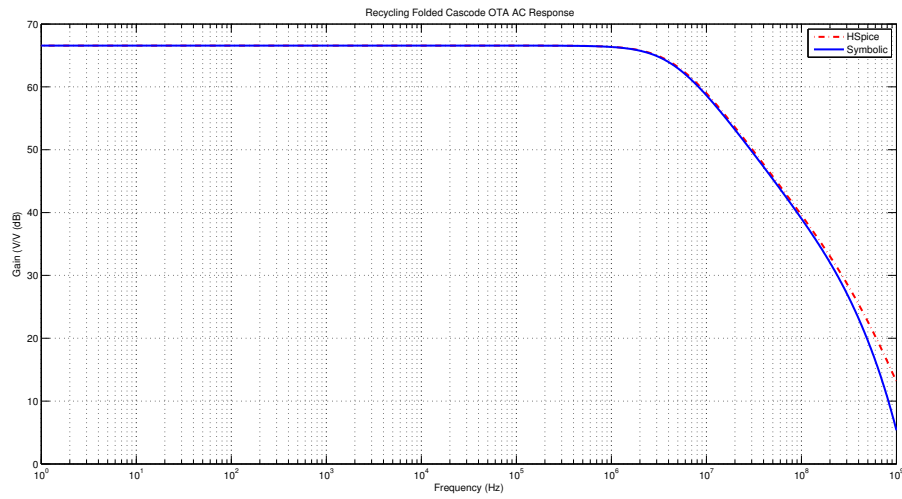


Figure 3.12:  $H(s)$  comparison between HSPICE<sup>TM</sup> (dashed) and Graph Based Tool (solid) for the RFC OTA [49] topology.

## Chapter 4

# Constrained Optimization

This chapter focuses on the constrained optimization problem and looks into different methods for solving it. Constrained optimization is used in order to perform the proposed non Monte Carlo method since the problem of finding lower and upper bounds of the magnitude and phase responses can be formulated in a nonlinear constrained optimization.

Given a  $s$ -expanded transfer function, i.e.,

$$H(\omega) = \frac{\sum_{i=0}^m a_i(p_1, \dots, p_m) s^i}{\sum_{j=0}^n b_j(p_1, \dots, p_n) s^j} \quad (4.1)$$

where the coefficients  $a_i(p_1, \dots, p_m)$  and  $b_j(p_1, \dots, p_n)$  are obtained by the method explained in the previous chapter. Notice that  $H(s, p_1, \dots, p_m)$  is a nonlinear function of  $p_1, \dots, p_m$ .

In this Thesis it is assumed that each parameter  $p_i$  is a random variable with a variational range. Since  $s = j\omega$ , instead of getting a nominal transfer function

$$H(s) = H(j\omega) = H(j\omega)e^{j\theta(\omega)} \quad (4.2)$$

one should obtain a variational transfer function with bounded magnitude and phase regions, i.e.,

$$H_l(\omega) \leq H(\omega) \leq H_u(\omega) \quad (4.3)$$

$$\Theta_l(\omega) \leq \Theta(\omega) \leq \Theta_u(\omega) \quad (4.4)$$

Where  $H_l(\omega)$  and  $H_u(\omega)$  are the lower and upper bounds of the magnitude respectively, and  $\theta_l(\omega)$  and  $\theta_u(\omega)$  are the lower and upper bounds of the phase respectively. The evaluation of the transfer function described by (4.2) gives a complex valued result, where the magnitude  $H(\omega) = |H(j\omega)|$  and the phase angle  $\theta(\omega) = \angle H(j\omega)$  are real values. The goal of the proposed frequency response bound analysis is to derive the bounds of magnitude and phase for  $H(j\omega)$ , such that one can obtain (4.3) and (4.4).

Hence, in the presence of process variations, the signal is also perturbed from its nominal behavior, and are usually bounded between its minimum and maximum limits, as explained in Fig. 4.1

To find the bounds, one should formulate the bound computing problem into a nonlinear constrained optimization problem. To obtain the two performance bounds for magnitude or phase at one frequency point, two evaluation processes, or optimization runs, of the transfer function are needed: one for  $H(\omega)$ , and the other for  $\theta(\omega)$ . We use

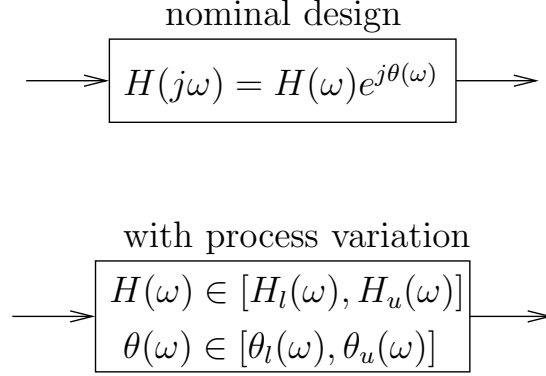


Figure 4.1: Circuit system with process variation.

lower bound of the magnitude response  $|H(j\omega)|$  at frequency  $\omega$ . The magnitude of the transfer function, which can be evaluated from the available symbolic transfer function, is used as the nonlinear objective function to be minimized:

$$\begin{aligned}
 &\text{minimize} && |H(j\omega, \mathbf{x})| \\
 &\text{subject to} && \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u
 \end{aligned} \tag{4.5}$$

which is a box constrained optimization problem and where  $\mathbf{x} = [p_1, \dots, p_m]$  represents circuit parameter variable vector, which are subjected to the optimization constraints  $[\mathbf{x}_l, \mathbf{x}_u]$ . In circuit design, these constraints are supplied by foundries and cell library vendors. Hence, after (4.5) is solved by an optimization engine, the lower bound of magnitude response at  $\omega$ , i.e.,  $|H_1(j\omega)|$ , is returned and a parameter set at which the minimum is attained will also be saved as a by-product. Fig. 4.2 and Algorithm 5 summarizes the flow of the performance bound calculation.

Unlike unconstrained optimization, constrained optimization's goal isn't the global

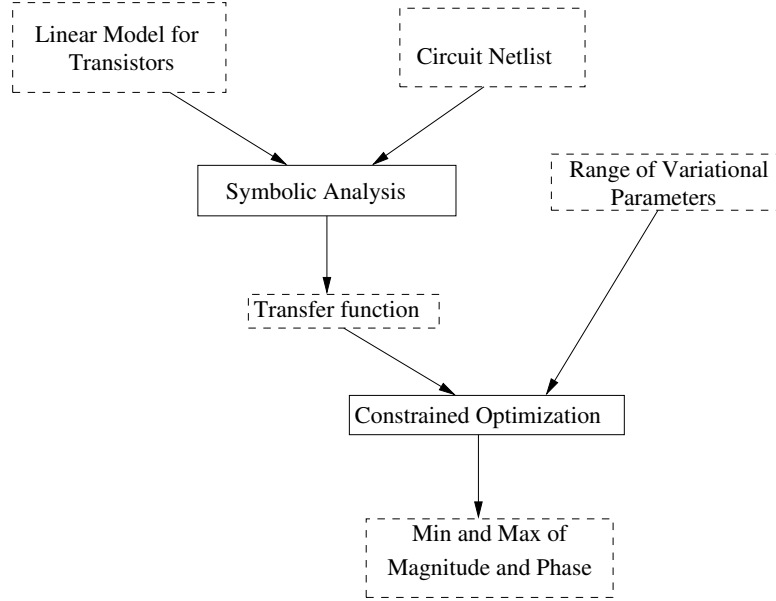


Figure 4.2: Flowchart of frequency domain performance bound calculation.

optima [51, 52].

## 4.1 A simple example in MATLAB

This section describes a specific example to show how to approach this constrained optimization problem. In Fig. 4.3 is shown the circuit diagram of a RLC ladder circuit,

---

**Algorithm 5** Calculation of frequency response bounds using the graph based method presented in Chapter 2 and symb2 and constrained optimization

---

- 1: Parse circuit netlist.
  - 2: Set bounds on process variation affected parameters.
  - 3: Run the graph based algorithm to generate the symbolic expression like the transfer function.
  - 4: **for** All frequencies  $\omega_i$  **do**
  - 5:     Constrained optimization to find magnitude bounds on  $\omega_i$ .
-

where the three components are all in series with the voltage source.

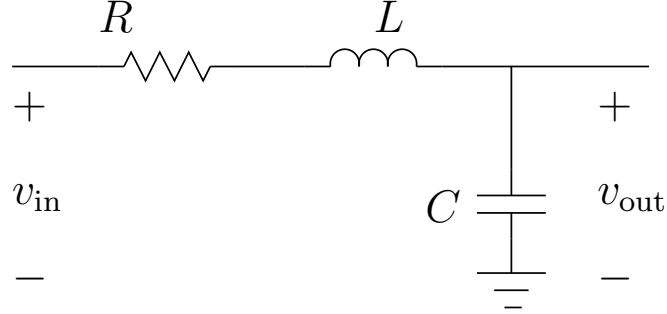


Figure 4.3: A RLC ladder circuit

In frequency domain, capacitors and inductors are analyzed as complex impedances, and one can assume the voltage on the capacitors are taken for observations, then the transfer function of this circuit is:

$$H(j\omega) = \frac{V_{out}}{V_{in}} = \frac{\frac{1}{j\omega C}}{R + j\omega L + \frac{1}{j\omega C}} \quad (4.6)$$

For this example, it can be assumed that the capacitor and inductor are under process variation. For their nominal parameters  $C = 1\mu F$  and  $L = 1\mu H$ , and assuming a variation of 20% on the inductor and capacitor, i.e.,  $C \in [0.8, 1.2]\mu F$  and  $L \in [0.8, 1.2]\mu H$ , one obtain the results shown in Fig. 4.4 are plotted as a solid line the nominal curve and as dashed curves the lower bound in green and the upper bound in magenta. In the same figure, three snapshots of the transfer function at different frequency points are shown. Each snapshot show the impact of the variations of each element (the inductor and capacitor in this specific case) in the magnitude  $H(\omega)$ .

For this example an iterative method called active set was used. Active-Set methods

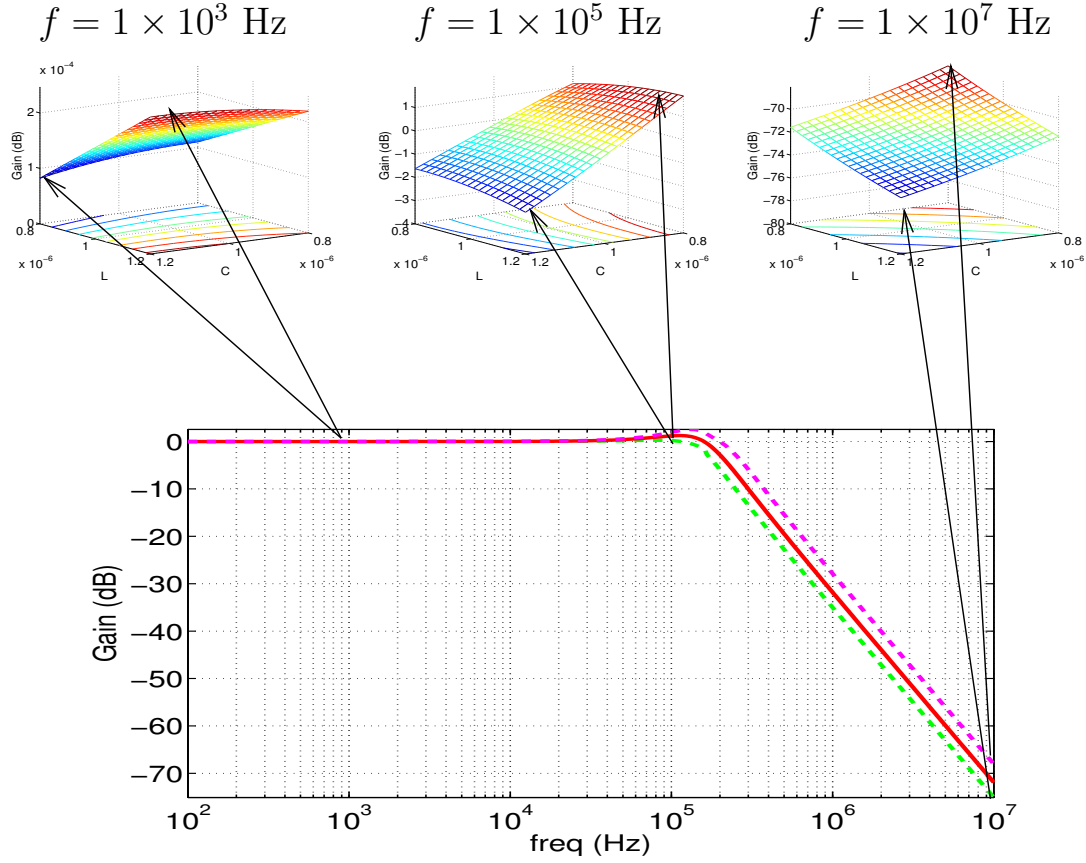


Figure 4.4: Frequency response of the RLC circuit in Fig. 4.3. Solid curve is the magnitude response with nominal parameters, while the two dashed curves are lower and upper bounds due to process variation. The three surfaces at the top, with  $L$  and  $C$  as  $x$ -axis and  $y$ -axis accordingly and magnitude as  $z$ -axis, illustrate the variations of magnitude at three sampling frequencies.

are two-phase iterative methods that provide an estimate of the active set (active set is the set of constraints that are satisfied with equality) at the solution. In the first phase, the objective is ignored while a feasible point is found for the constraints. In the

second phase, the objective is minimized while feasibility is maintained. In the second phase, starting from a feasible initial point  $x_0$ , the method computes a sequence of feasible iterates  $\{x_k\}$  such that  $x_{k+1} = x_k + \alpha_k p_k$  and  $H(x_{k+1}) \leq H(x_k)$  via methods like quadratic programming, where  $p_k$  is a nonzero search and  $\alpha_k$  is a nonnegative step length. This iterative search procedure in the constrained minimization is illustrated in Fig. 4.5 for a specific frequency.

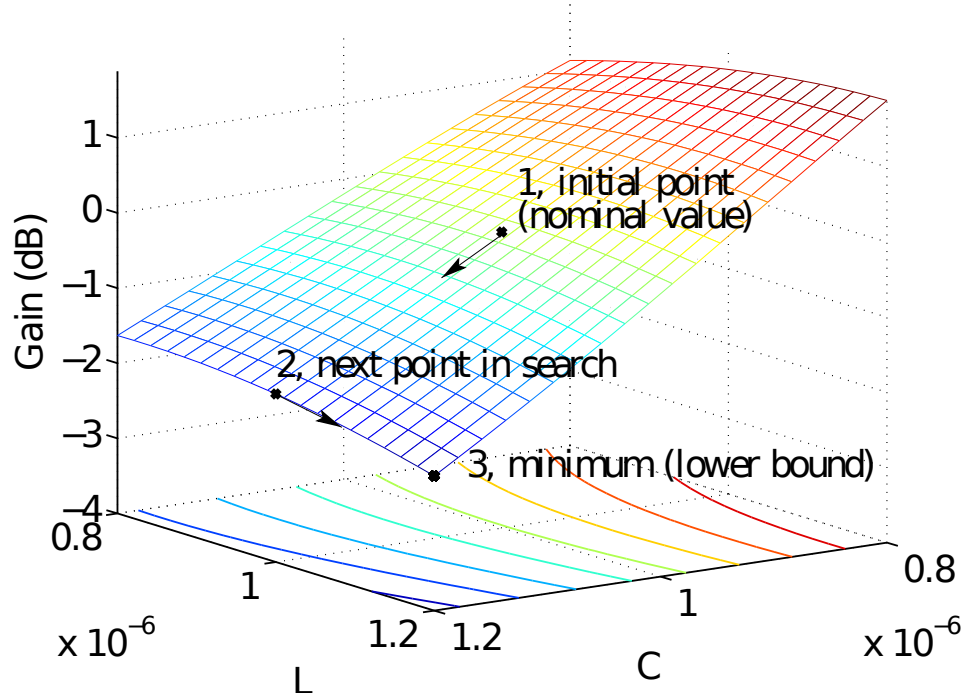


Figure 4.5: Optimization space searching for the RLC circuit in Fig. 4.3, where both  $C$  and  $L$  have 20% variation in this illustration. This surface shows the magnitude variations at frequency  $f = 10^6 \text{Hz}$ .



## 4.2 Line Search

A line search algorithm searches for the decrease in  $f$  in a descent direction, using the Armijo rule [53, 40, 41] for step size control. Most line searches used in practice are inexact: the step length is chosen to approximately minimize  $f$  along the ray  $\{x + \lambda\Delta x | \lambda \geq 0\}$  where  $\Delta x$  is called the *descent direction* and the point  $x + \lambda\Delta x$  is called  $x_+$ , or even to just reduce  $f$  'enough' [40]. Once given the current point  $x_c$  and the descent direction  $\Delta x$ , we look for  $\lambda$  such that:

$$f(x_c + \lambda\Delta x) \leq f(x_c) \quad (4.7)$$

However, if the decreasing achieved at the inequality above for some  $\lambda$  is too small, it is not possible to guarantee convergence to a local minimum. So in order to avoid this, it is needed that  $\lambda$  satisfies the *Armijo rule* (*sufficient decrease*) given by:

$$f(x_c + \lambda\Delta x) \leq f(x_c) + \lambda\gamma\nabla f(x_c)^T\Delta x \quad (4.8)$$

where  $\gamma \in (0, 1)$  [41]. This is shown in Fig. 4.6

In Fig. 4.6 as we can see the interval where the condition (4.8) holds is  $[0, \lambda_{max}]$  so we can rewrite it as:

$$f(x_+) \leq f(x_c) + \gamma\nabla f(x_c)^T(x_+ - x_c) \quad (4.9)$$

The Armijo condition [54] to accept a trial point  $x_+$  is

$$f(x_+) \leq \max_{0 \leq j \leq \min\{c, M-1\}} f(x_{c-j}) + \gamma\nabla f(x_c)^T(x_+ - x_c) \quad (4.10)$$

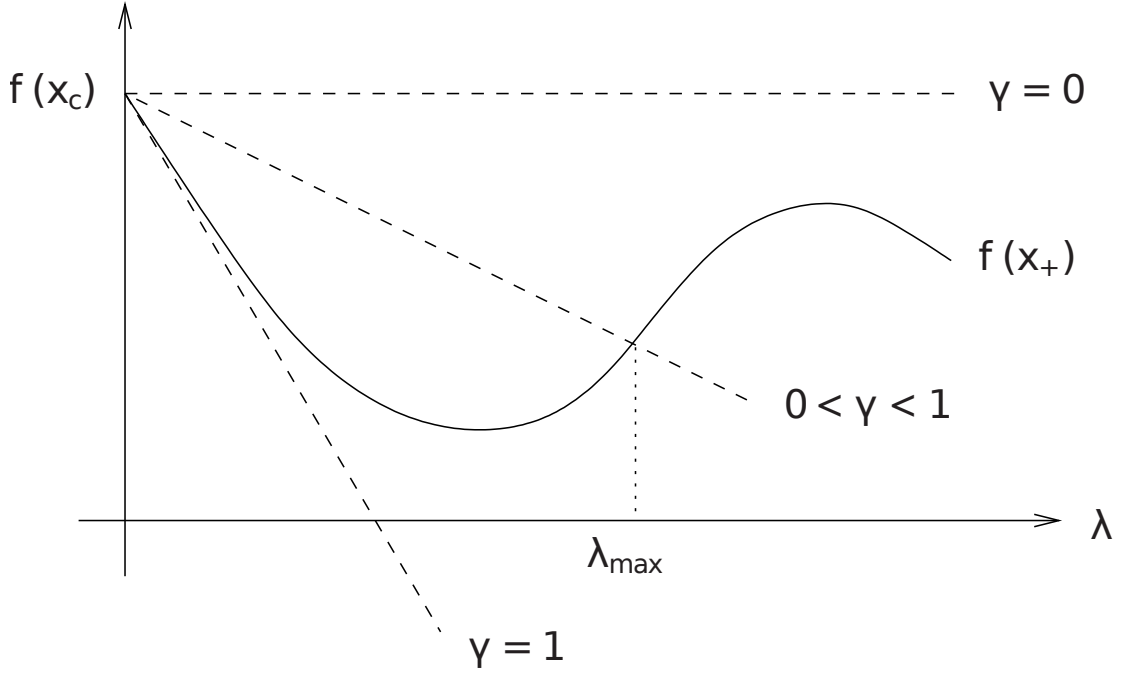


Figure 4.6: Line Search representation

where  $M$  is an integer greater than zero. If  $x_+$  is rejected, the steplength is redefined as

$$\lambda = \max \{ \sigma_1 \lambda, \min \{ \sigma_2 \lambda, \lambda^* \} \}, \quad (4.11)$$

where  $\lambda^*$  minimizes a quadratic model. This strategy of repeatedly testing for sufficient decrease and reducing the stepsize if the test is failed is called backtracking for obvious reasons. The methods described below are based on backtracking line search.

### 4.3 Projected Gradient Method

The gradient projection algorithm [38, 55] is the natural extension of the *steepest descent algorithm* [41], used in unconstrained optimization, to bound constrained problems. The

method is iterative as the Active-Set method explained before in Section 4.1 . This method is presented in [41]. The line search method applied here is backtracking line search explained in section 4.2. Given a current iterate  $x_c$  the new iterate is

$$x_+ = P(x_c - \lambda \nabla f(x_c)) \quad (4.12)$$

where the gradient  $\nabla f(x)$  is defined by

$$\begin{aligned} \nabla f(x) &= \left( \frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right) \\ \text{for } x &= (x_1, \dots, x_n) \end{aligned} \quad (4.13)$$

where  $\lambda$  is a steplength parameter given by the Armijo rule, where the Armijo rule is a *line search* in which one searches on a ray from  $x_c$  in a direction in which  $f$  is locally decreasing [54]. In order to implement any line search scheme, we must specify what we mean by sufficient decrease. For bound constrained problems the *sufficient decrease* condition for line searches will be defined as

$$f(x(\lambda)) - f(x) \leq \frac{-\alpha}{\lambda} \|x - x(\lambda)\|^2 \quad (4.14)$$

where

$$x(\lambda) = P(x - \lambda \nabla f(x)) \text{ for } \lambda \geq 0 \quad (4.15)$$

It is important to choose the steplength  $\lambda$ . One way to do this is to let  $\lambda = \beta^m$ , where  $\beta \in (0, 1)$  and  $m \geq 0$  which is the smallest nonnegative integer such that there is *sufficient decrease* in  $f$ . For the termination condition first is necessary to define what

is an active set and an inactive set. The set of constraints is called the *feasible set* ( $\Omega$ ) and a point in this set is called *feasible point*. Because the *feasible set* is compact there is always a solution for this minimization problem. The  $i$ th constraint is *active* at  $x \in \Omega$  if either  $x_i = L_i$  or  $x_i = U_i$ . If the  $i$ th constraint is not *active* it will be called *inactive*. One can write  $A(x)$  and  $I(x)$  the active and inactive set respectively, where an *active set* is the set of indexes  $i$  such that the  $i$ th constraint is active and the *inactive set* is the set of indexes  $i$  such that the  $i$ th constraint is inactive.

Therefore,

$$r_0 = \|x_0 - x_0(1)\| \quad (4.16)$$

and the relative and absolute tolerances  $\tau_r$  and  $\tau_a$  then the termination criterion is give by

$$\|x - x(1)\| \leq \tau_a + \tau_r r_0 \quad (4.17)$$

The algorithm for this optimization method is shown in the Algorithm 6. It starts with  $x_0 \in \Omega$ , and uses a sufficient decrease parameter  $\gamma \in (0, 1)$  and safeguarding parameters  $0 \leq \sigma_1 \leq \sigma_2 \leq 1$ . Initially,  $\alpha_0 = 1/\|f(x_0)\|^2$ . Given  $x_c$  and  $\alpha_c \geq 0$  the algorithm shows how to obtain  $x_{c+1}$  and  $\alpha_{k+1}$ , and when to finish the optimization.

## 4.4 Spectral Projected Gradient Method

This method has improved the choosing of the steplength. In this methods the choice of the septlength requires little computational work and greatly speeds up the convergence

**Algorithm 6** Projected Gradient Method

---

```

1: if  $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$ , i.e.,  $x_c$  is stationary then
2:   Stop
3: Compute  $d_c = P(x_c - \alpha_c \nabla f(x_c)) - x_c$ .
4: Set  $\lambda \leftarrow 1$ 
5: Set  $x_+ = x_c + \lambda d_c$ 
6: if  $f(x_+) \leq f(x_c) + \gamma \langle x_+ - x_c, \nabla f(x_c) \rangle$  then
7:    $\lambda_c = \lambda$ ,  $x_{c+1} = x_+$ ,  $s_k = x_{c+1} - x_c$ ,  $y_k = \nabla f(x_{c+1})$ 
8:   Go to step 12
9: else
10:  Define  $\lambda_{new} \in [\sigma_1 \lambda, \sigma_2 \lambda]$  and  $\lambda \leftarrow \lambda_{new}$ 
11:  Go to Step 5
12: Compute the trial step length  $\alpha_{c+1} = \alpha_c \|f(x_c)\|^2 / \|f(x_{c+1})\|^2$ .

```

---

of gradient methods [56]. Unlike the projected gradient method explained in section 4.3, Spectral Projected Gradient Method [38, 39] is more related to the quasi-Newton family of methods [53] through an approximated secant equation. The main idea behind the spectral choice of steplength is that the steepest descent method is very slow but it can be accelerated taking, instead of the stepsize that comes from the minimization of the function along the gradient of the current iteration, the one that comes from the one-dimensional minimization at the previous step.

The point in the first iteration of this method should be a feasible point, i.e. the algorithm starts with  $x_0 \in \Omega$  and uses an integer  $M \geq 1$ , a small parameter  $\alpha_{min} \geq 0$ ; a large parameter  $\alpha_{max} \geq \alpha_{min}$ ; a sufficient decrease parameter  $\gamma \in (0, 1)$ , and safeguarding parameters  $0 \leq \sigma_1 \leq \sigma_2 \leq 1$ .  $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$ . The algorithm is shown in Algorithm 7. In this algorithm  $\lambda_{new}$  uses one-dimensional quadratic interpolation and it is safeguarded taking  $\lambda \leftarrow \lambda/2$  when the minimum of the one-dimensional quadratic

**Algorithm 7** Spectral Projected Gradient Method

---

```

1: if  $\|P(x_c - \nabla f(x_c)) - x_c\| = 0$ , i.e.,  $x_c$  is stationary then
2:   Stop
3: Compute  $d_c = P(x_c - \alpha_c g(x_c)) - x_c$ .
4: Set  $\lambda \leftarrow 1$ 
5: Set  $x_+ = x_c + \lambda \Delta f(x_c)$ 
6: if  $f(x_+) \leq \max_{0 \leq j \leq \min\{c, M-1\}} f(x_{c-j}) + \gamma \lambda \langle d_c, \nabla f(x_c) \rangle$  then
7:   Define  $\lambda_c = \lambda$ ,  $x_{c+1} = x_+$ ,  $s_c = x_{c+1} - x_c$ ,  $y_c = \nabla f(x_{c+1}) - \nabla f(x_c)$ 
8:   Got to Step 11
9: else
10:  Set  $\lambda_{new} \in [\sigma_1 \lambda, \sigma_2 \lambda]$ ,  $\lambda \leftarrow \lambda_{new}$  and go to Step 5
11: Compute  $b_c = \langle s_c, y_c \rangle$ .
12: if  $b_k \leq 0$  then
13:    $\alpha_{c+1} = \alpha_{max}$ .
14: else
15:    $\alpha_c = \langle s_c, s_c \rangle$  and  $\alpha_{c+1} = \min\{\alpha_{max}, \max\{\alpha_{min}, \alpha_c/b_c\}\}$ 

```

---

lies outside  $[0.1, 0.9\lambda]$ . Notice that the line search condition in step 6 of Algorithm 7 guarantee that the sequence  $\{x_c\}$  remains in  $\Omega_0 = \{x \in \Omega : f(x) \leq f(x_0)\}$ .

## 4.5 Remarks

Both of the algorithms, projected gradient method and spectral projected gradient method, start at  $x_0 \in \Omega$  and use as search direction the internal projected gradient direction. In case of rejection of the first trial point, the next ones are computed along the same line. Also for both algorithms, the calculation of  $\lambda_{new}$  uses a one dimensional quadratic interpolation. The computational work, for both algorithms, involves a projection on the convex set  $\Omega$ , a function evaluation ( $f(x)$ ) and a gradient evaluation per iteration ( $\nabla f(x)$ )

## Chapter 5

# Experimental Results

The circuits used to prove the usefulness of the proposed approach results are shown in Fig. 5.1, a Differential Pair and two and three stages operational transconductance amplifiers. As already said in Chapter 4 in Fig. 4.2 and in Algorithm 5 for the bound analysis approach one first parse the circuit netlist given, later the bound on process variations of the affected parameter are set and the symbolic expression like the transfer function presented in (4.1) is generated. Given the transfer function for all desired frequencies (this is set by the user) constrained optimization is used to find magnitude bounds. Projected gradient method and Spectral Projected Gradient Method are programmed in C compiled in an Ubuntu Linux Environment with the GNU C compiler gcc-4.6.1 with 4GB RAM. Active-Set method is part of the matlab optimization tool. The experiments presented in this chapter were run in a Intel Core i3

The results of the optimization problem are shown in this chapter. Results for the differential pair are shown in Figures 5.2a and 5.2b. Results for the 3 stages OTA are shown in Figures 5.3a and 5.3b. In Table 5.1 a time comparative table is shown between the methods.

Name	PGRAD	SPG	ACTIVE SET <sup>1</sup>	HSPICE <sup>TM</sup>
Differential Pair	15.312ms	16.363ms	4.524s	1.716s
2 Stages OTA	44.559ms	91.936ms	4.833s	81.65ms
3 Stages OTA	38.315ms	39.233ms	5.2855s	2.163s

Table 5.1: Time comparison between the methods used

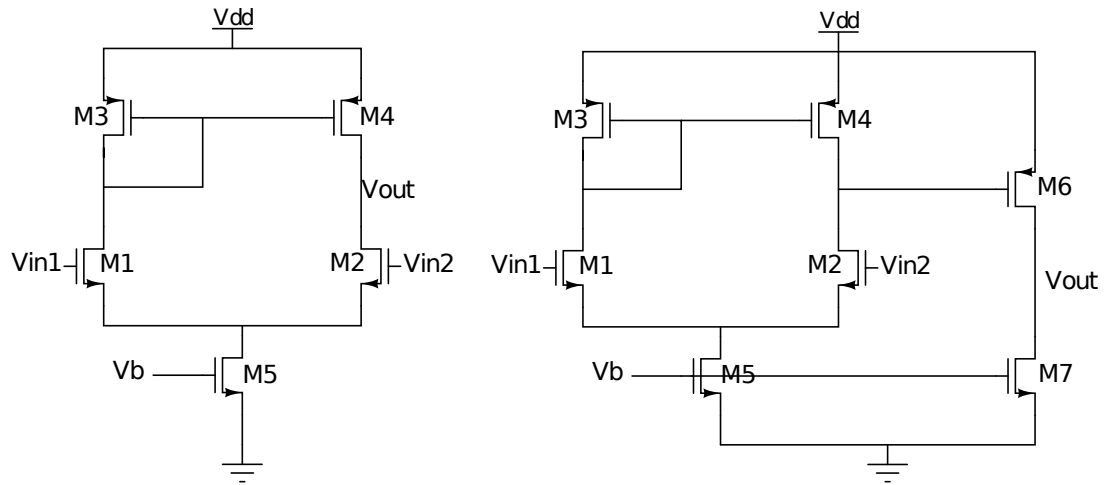
It is important to note that the circuits used in order to compare the computed results were small signal representations of the mentioned circuits since the model used in the graph-based tool to perform symbolic analysis is not accurate enough. Making the MOSFET model more accurate would increase the computational cost of a symbolic tool. Another drawback of using symbolic analysis for variational analysis in MOSFET based circuits is that as they are expressed by a small signal representation in the symbolic analysis, it is very hard to choose which element(element) in the expression given by the symbolic tool is(are) the one(s) which one has to choose as the variational element for any bound analysis method, in this case constrained optimization, i.e,  $g_m$ ,  $C_{gd}$ ,  $C_{gs}$ ,  $g_o$ , etc., in the MOSFET in order to do the same variation to the MOSFET variational

---

<sup>1</sup>Matlab optimization tool.

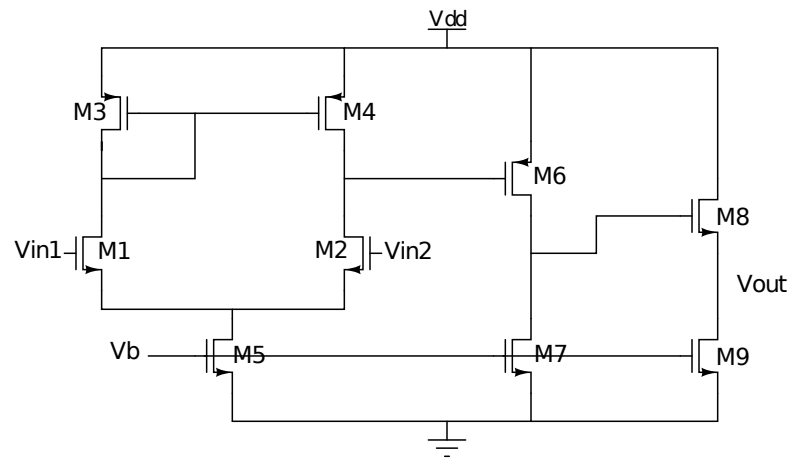


analysis needed in HSPICE<sup>TM</sup>, i.e.,  $L$ ,  $W$ ,  $V_T$ , etc. Nevertheless this last issue, the approach showed good time results, eventhough one can't see any determined behavior in the time comparison between the differente methods used in this Thesis, since they are small despite symbolic analysis is used. And observation is that variational analysis based on symbolic analysis would be better for linearized networks.



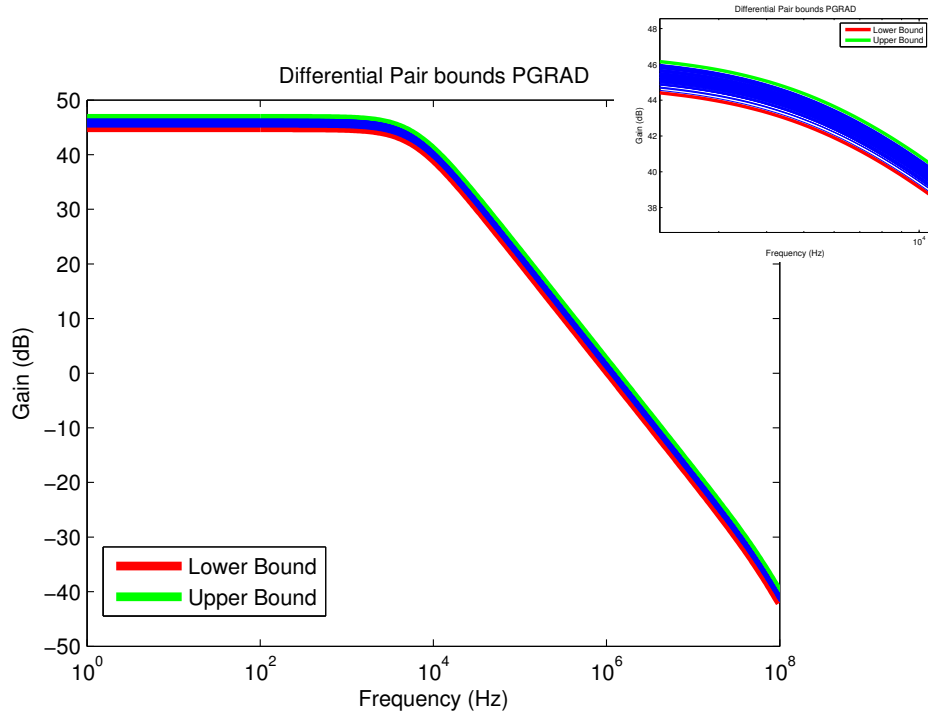
(a) Differential Pair

(b) 3 stages OTA

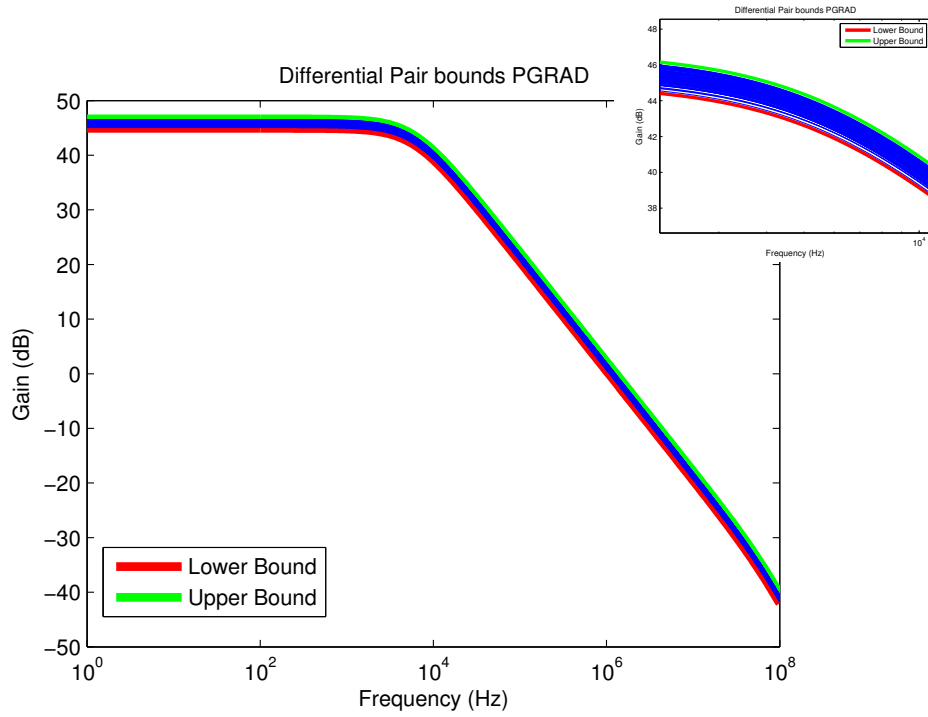


(c) 3 stages OTA

Figure 5.1: Circuits used for the simulations

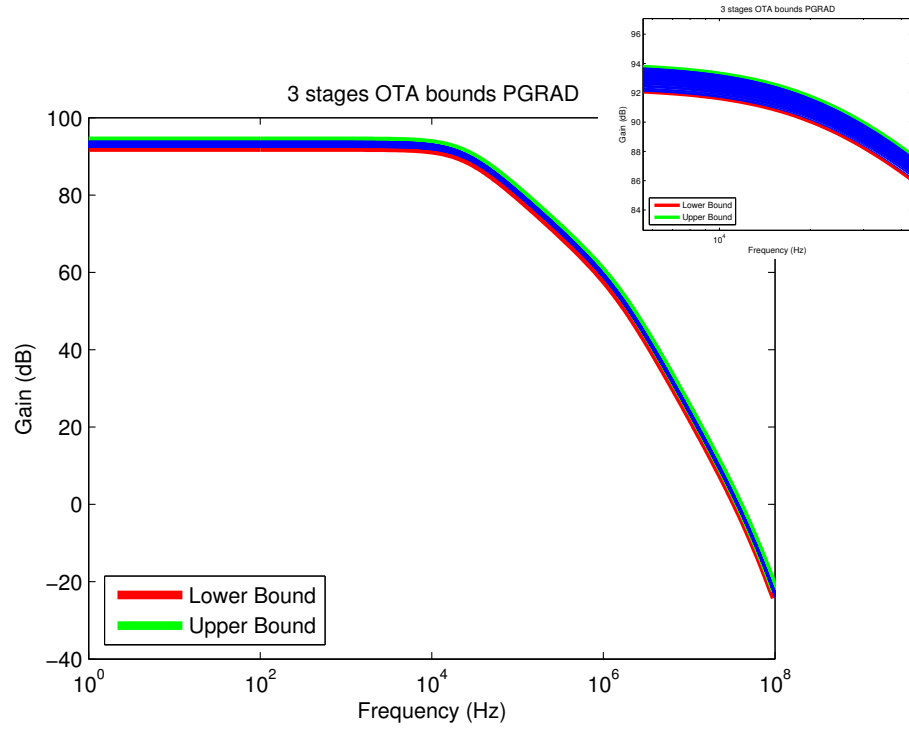


(a) Differential Pair Bound Constrains obtained with PG Method

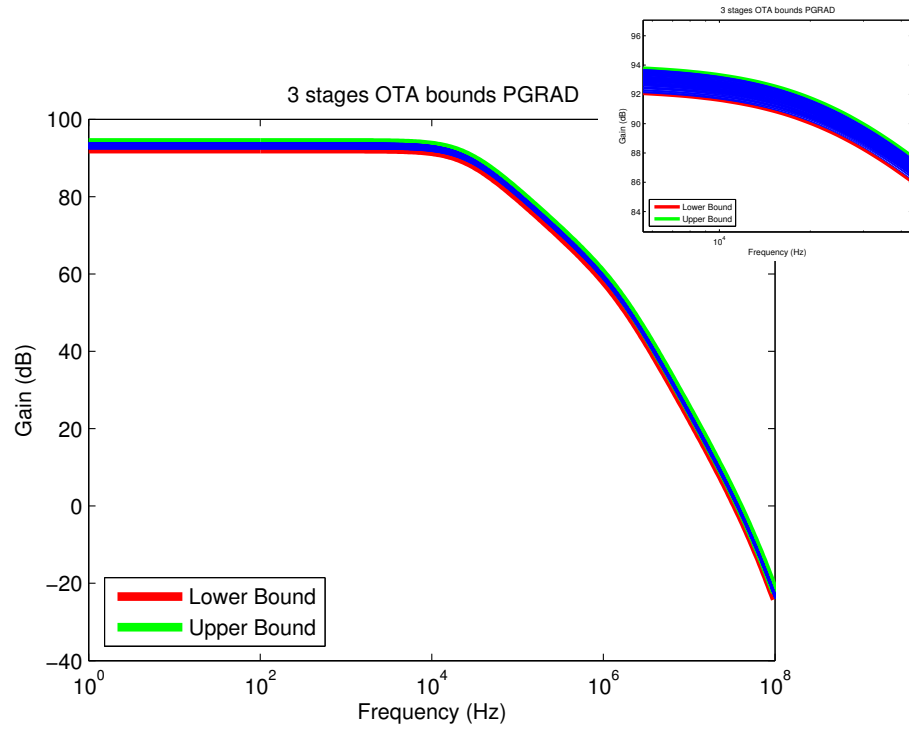


(b) Differential Pair Bound Constrains obtained with SPG Method

Figure 5.2: Results for the differential pair



(a) 3 Stage OTA Bound Constrains obtained with PG Method



(b) 3 stage OTA Bound Constrains obtained with SPG Method

Figure 5.3: Results for the 3 stage OTA

## Chapter 6

# Conclusions

In this work a new approach to obtain the bounds of a circuit was presented based on constrained optimization using line search methods such as projected gradient method, spectral projected gradient method and active set method. A new graph-based method was used in order to perform symbolic analysis that can run with the same netlist for input that HSPICE<sup>TM</sup> handles avoiding the need for intermediary processing steps. With this approach it is possible to fully define the circuit variables in terms of node voltages or current branches. It was proved that the bound analysis presented here was successful for the examples in spite of the small signal models that aren't accurate. Increase in accuracy would be an increase in computational work.

The running time of the bound analysis was comparable with HSPICE<sup>TM</sup> nonetheless a symbolic approach is used. It is important to note that symbolic analysis can be used

as a tool in order to understand a circuit qualitatively more than quantitatively. This important characteristic make symbolic analysis not the best tool for exact variational analysis but it helps to understand the behavior of the circuit of interest.

As future work, other variational analysis approaches should be studied to include nonlinear circuits like CMOS analog ICs, and more analysis should be done to associate parameter design variables as width/length of the MOSFETs, to their circuit equivalents.

# List of Publications

- A.A. Palma-Rodriguez, E. Tlelo-Cuautle, S. Rodriguez-Chavez, and S.X. Tan.  
DDD-based symbolic sensitivity analysis of active filters. In *Devices, Circuits and Systems (ICCDCS), 2012 8th International Caribbean Conference on*, pages 1 --4, march 2012.
- S. Rodriguez-Chavez, E. Tlelo-Cuautle, A.A. Palma-Rodriguez, and S.X.-D. Tan.  
Symbolic DDD-based tool for the computation of noise in CMOS analog circuits.  
In *Devices, Circuits and Systems (ICCDCS), 2012 8th International Caribbean Conference on*, pages 1 --4, march 2012.
- S. Rodriguez-Chavez, A.A. Palma-Rodriguez, E. Tlelo-Cuautle, S. X.-D. Tan,  
*Graph-based symbolic and symbolic sensitivity analysis of analog integrated circuits,*  
*in Analog/RF and Mixed-Signal Circuit Systematic Design*, M. Fakhfakh, E. Tlelo-Cuautle, R. Castro-Lpez (Eds.), Springer, 2012. In press

# List of Figures

2.1	Graph representation of Equation (2.1)	12
2.2	Determinant graph	18
2.3	Expanded graph	20
3.1	Symbolic Tool Flow	23
3.2	Module for Symbolic Derivative	24
3.3	Pathological element symbol for a) Nullator and b) Norator.	27
3.4	Nullor representation of the MOSFET	27
3.5	Nullor representation of a three stage OTA	28
3.6	Nullor representation of non-NA compatible elements	30
3.7	Nullor Equivalent of the MOSFET Differential Pair	32
3.8	Nullor based (a) Voltage Follower and (b) Current Follower	33
3.9	Common Source Small Signal Equivalent	35
3.10	Nullor equivalents of the amplifier circuits.	37



3.11	H(s) comparison between DDD (SCAD3) [43] (dashed), HSPICE <sup>TM</sup> (dots) and Graph Based Tool (solid) for the differential pair topology. . . . .	39
3.12	H(s) comparison between HSPICE <sup>TM</sup> (dashed) and Graph Based Tool (solid) for the RFC OTA [49] topology. . . . .	40
4.1	Circuit system with process variation. . . . .	43
4.2	Flowchart of frequency domain performance bound calculation. . . . .	44
4.3	A RLC ladder circuit . . . . .	45
4.4	Frequency response of the RLC circuit in Fig. 4.3. Solid curve is the magnitude response with nominal parameters, while the two dashed curves are lower and upper bounds due to process variation. The three surfaces at the top, with L and C as $x$ -axis and $y$ -axis accordingly and magnitude as $z$ -axis, illustrate the variations of magnitude at three sampling frequencies.	46
4.5	Optimization space searching for the RLC circuit in Fig. 4.3, where both $C$ and $L$ have 20% variation in this illustration. This surface shows the magnitude variations at frequency $f = 10^6\text{Hz}$ . . . . .	47
4.6	Line Search representation . . . . .	49
5.1	Circuits used for the simulations . . . . .	57
5.2	Results for the differential pair . . . . .	58
5.3	Results for the 3 stage OTA . . . . .	59

# List of Tables

3.1	Elements Tables . . . . .	25
3.2	NA Formulation . . . . .	35
3.3	Symbolic Formulation and Numerical Evaluation of $D(s)$ , $N(s)$ and $H(s)$	38
5.1	Time comparison between the methods used . . . . .	55

# Bibliography

- [1] Ying Liu, L.T. Pileggi, and A.J. Strojwas. Model order-reduction of rc(l) interconnect including variational analysis. In *Design Automation Conference, 1999. Proceedings. 36th*, pages 201 --206, 1999.
- [2] Rob A. Rutenbar. Next-generation design and EDA challenges: Small physics, big systems, and tall tool-chains. In *Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific*, page vii, jan. 2007.
- [3] S.R. Nassif. Model to hardware matching for nano-meter scale technologies. In *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, pages 28 --31, sept. 2007.
- [4] Jaeha Kim, K.D. Jones, and M.A. Horowitz. Fast, non-Monte-Carlo estimation of transient performance variation due to device mismatch. In *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pages 440 --443, june 2007.

- [5] H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki. Challenge: variability characterization and modeling for 65- to 90-nm processes. In *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, pages 593 -- 599, sept. 2005.
- [6] A. Singhee and R.A. Rutenbar. Why quasi-Monte Carlo is better than Monte Carlo or latin hypercube sampling for statistical circuit analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(11):1763 --1776, nov. 2010.
- [7] C. Kemp. Monte Carlo analysis and design-application to a crystal-controlled oscillator. *Communication Technology, IEEE Transactions on*, 14(3):329 --333, june 1966.
- [8] Pasi E. Lassila and Jorma T. Virtamo. Nearly optimal importance sampling for Monte Carlo simulation of loss systems. *ACM Trans. Model. Comput. Simul.*, 10(4):326--347, October 2000.
- [9] T. Kida, Y. Tsukamoto, and Y. Kihara. Optimization of importance sampling monte carlo using consecutive mean-shift method and its application to sram dynamic stability analysis. In *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pages 572 --579, march 2012.
- [10] A.A. Bayrakci, A. Demir, and S. Tasiran. Fast monte carlo estimation of timing yield with importance sampling and transistor-level circuit simulation. *Computer-*

- Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(9):1328--1341, sept. 2010.
- [11] P. Lassila, J. Karvo, and J. Virtamo. Efficient importance sampling for Monte Carlo simulation of multicast networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 432--439 vol.1, 2001.
- [12] Yin Jinhang, Lu Wenxi, Xin Xin, and Zhang Lei. Application of monte carlo sampling and latin hypercube sampling methods in pumping schedule design during establishing surrogate model. In *Water Resource and Environmental Protection (ISWREP), 2011 International Symposium on*, volume 1, pages 212--215, may 2011.
- [13] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):pp. 55--61, 2000.
- [14] Mansour Keramat and Richard Kielbasa. Latin hypercube sampling Monte Carlo estimation of average quality index for integrated circuits. *Analog Integrated Circuits And Signal Processing*, 14(1/2):131--142, 1997.
- [15] Mohd Azdi Maasar, Noraimi Azlin Mohd Nordin, Marlyn Anthonyrajah, Wan Murni Wan Zainodin, and Amani Mohammad Yamin. Monte carlo amp; quasi-

- monte carlo approach in option pricing. In *Humanities, Science and Engineering Research (SHUSER), 2012 IEEE Symposium on*, pages 1401 --1405, june 2012.
- [16] Hui Zhang and Chongzhao Han. A new quasi-monte carlo filtering algorithm based on number theoretical method. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 2186 --2191, june 2010.
- [17] Ruijing Shen, Sheldon X.-D. Tan, and Hao Yu. *Statistical performance analysis and modeling techniques for nanometer VLSI designs*. Springer, 2012.
- [18] L.V. Kolev, V.M. Mladenov, and S.S. Vladov. Interval mathematics algorithms for tolerance analysis. *Circuits and Systems, IEEE Transactions on*, 35(8):967 --975, aug 1988.
- [19] Wei Tian, Xie-Ting Ling, and Ruey-Wen Liu. Novel methods for circuit worst-case tolerance analysis. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 43(4):272 --278, apr 1996.
- [20] M.W. Tian and R.C.-J. Shi. Worst case tolerance analysis of linear analog circuits using sensitivity bands. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 47(8):1138 --1145, aug 2000.
- [21] Liuxi Qian, Dian Zhou, Sheng-Guo Wang, and Xuan Zeng. Worst case analysis of linear analog circuit performance based on Kharitonov's rectangle. In *Solid-*

- State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE International Conference on*, pages 800 --802, nov. 2010.
- [22] Xue-Xin Liu, S.X.-D. Tan, Zhigang Hao, and Guoyong Shi. Time-domain performance bound analysis of analog circuits considering process variations. In *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pages 535 --540, 30 2012-feb. 2 2012.
- [23] F. Yuan. Efficient non-Monte Carlo method for statistical analysis of periodically switched linear circuits in frequency domain. *Circuits, Devices and Systems, IEE Proceedings -*, 150(5):423--8, oct. 2003.
- [24] G. Gielen, P. Wambacq, and W.M. Sansen. Symbolic analysis methods and applications for analog circuits: a tutorial overview. *Proceedings of the IEEE*, 82(2):287 --304, feb 1994.
- [25] W. Sansen, G. Glelen, and H. Walscharts. A symbolic simulator for analog circuits. In *Solid-State Circuits Conference, 1989. Digest of Technical Papers. 36th ISSCC., 1989 IEEE International*, pages 204 --205, feb. 1989.
- [26] G.G.E. Gielen, H.C.C. Walscharts, and W.M.C. Sansen. ISAAC: a symbolic simulator for analog integrated circuits. *Solid-State Circuits, IEEE Journal of*, 24(6):1587 --1597, dec 1989.

- [27] A. Fernández F. V., Rodríguez-Vázquez and J. L. Huertas. A tool for symbolic analysis of analog integrated circuits including pole/zero extraction. In *Proc. 10th European Conf. on Circuit Theory and Design 2*, pages 752--761, September 1991.
- [28] F. V. Fernández, A. Rodríguez-Vázquez, and J. L. Huertas. Interactive AC modeling and characterization of analog circuits via symbolic analysis. *Analog Integrated Circuits and Signal Processing*, 1:183--208, 1991. 10.1007/BF00195622.
- [29] S.J. Seda, M.G.R. Degrauwe, and W. Fichtner. A symbolic analysis tool for analog circuit design automation. In *Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on*, pages 488 --491, nov 1988.
- [30] S.J. Seda, M.G.R. Degrauwe, and W. Fichtner. Lazy-expansion symbolic expression approximation in synap. In *Computer-Aided Design, 1992. ICCAD-92. Digest of Technical Papers., 1992 IEEE/ACM International Conference on*, pages 310 --317, nov, 1992.
- [31] S. Manetti. New approach to automatic symbolic analysis of electric circuits. *Circuits, Devices and Systems, IEE Proceedings G*, 138(1):22 --28, feb 1991.
- [32] G.M. Wierzba, A. Srivastava, V. Joshi, K.V. Noren, and J.A. Svoboda. SSPICE - a symbolic SPICE program for linear active circuits. In *Circuits and Systems, 1989., Proceedings of the 32nd Midwest Symposium on*, pages 1197 --1201 vol.2, aug 1989.



- [33] A. Konczykowska and M. Bon. Automated design software for switched-capacitor ic's with symbolic simulator scymbal. In *Proceedings of the 25th ACM/IEEE Design Automation Conference, DAC '88*, pages 363--368, Los Alamitos, CA, USA, 1988. IEEE Computer Society Press.
- [34] M.M. Hassoun and P.M. Lin. A new network approach to symbolic simulation of large-scale networks. In *Circuits and Systems, 1989., IEEE International Symposium on*, pages 806 --809 vol.2, may 1989.
- [35] L.P. Huelsman. Personal computer symbolic analysis program for undergraduate engineering courses. In *Circuits and Systems, 1989., IEEE International Symposium on*, pages 798 --801 vol.2, may 1989.
- [36] M. Fakhfakh, E. Tlelo-Cuautle, and F.V. Fernández. *Design of Analog Circuits through Symbolic Analysis*. Bentham Science Publishers Ltd., 2012.
- [37] Xiang-Dong Tan. *Symbolic analysis of large analog circuits with determinant decision diagrams*. PhD thesis, University of Iowa, May 1999.
- [38] Lenys Bello and Marcos Raydan. Convex constrained optimization for the seismic reflection tomography problem. *Journal of Applied Geophysics*, 62(2):158 -- 166, 2007.

- [39] Ernesto G. Birgin and José Mario Martínez. Large-Scale Active-Set Box-Constrained Optimization Method with Spectral Projected Gradients. *Computational Optimization and Applications*, 23:101--125, 2002. 10.1023/A:1019928808826.
- [40] Stephen Boyd and Lieven Vanderberghe. *Convex Optimization*. Cambridge University Press, 2009.
- [41] C.T. Kelley. *Iterative Methods for Optimization, Frontiers in Applied Mathematics*. SIAM, 1999.
- [42] Santiago Rodríguez-Chavez. Graph-based symbolic, symbolic noise and sensitivity analysis of CMOS analog integrated circuits. Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2012.
- [43] S. X.-D. Tan. Symbolic analysis by determinant decision diagrams and applications. In *Design of Analog Circuits through SYMBOLIC ANALYSIS*. Bentham Science Publishers Ltd., 2012.
- [44] C. Sanchez-Lopez, F.V. Fernandez, E. Tlelo-Cuautle, and S.X. Tan. Pathological element-based active device models and their application to symbolic analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 58(6):1382 --1395, june 2011.
- [45] D. Agnew. Efficient use of the Hessian matrix for circuit optimization. *Circuits and Systems, IEEE Transactions on*, 25(8):600 -- 608, aug 1978.

- [46] H. Carlin. Singular Network Elements. *Circuit Theory, IEEE Transactions on*, 11(1):67 -- 72, mar 1964.
- [47] Hung-Yu Wang, Wen-Chung Huang, and Nan-Hui Chiang. Symbolic nodal analysis of circuits using pathological elements. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 57(11):874 --877, nov. 2010.
- [48] S. Rodriguez-Chavez, E. Tlelo-Cuautle, A.A. Palma-Rodriguez, and S.X.-D. Tan. Symbolic DDD-based tool for the computation of noise in CMOS analog circuits. In *Devices, Circuits and Systems (ICCDSCS), 2012 8th International Caribbean Conference on*, pages 1 --4, march 2012.
- [49] I.G. Gómez. *Optimizacion multiobjetivo de circuitos electronicos aplicando algoritmos evolutivos*. PhD thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2012.
- [50] P. Patra, S. Kumaravel, and B. Venkatramani. An enhanced fully differential Recyclic Folded Cascode OTA. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1 --5, march 2012.
- [51] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods, nonlinear programming*. Springer, 2006.
- [52] E. K. P. Chong and S. H. Zak. *An introduction to optimization*. John Wiley and Sons, Inc. New York, 1996.

- [53] J. Frederic Bonnans, J. Charles Gilbert, Claude Lemarechal, and Claudia A. Sagastizabal. *Numerical Optimization, Theoretical and Practical Aspects*. Springer, 2000.
- [54] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1), 1966.
- [55] Ingrid Daubechies, Massimo Fornasier, and Ignace Loris. Accelerated Projected Gradient Method for Linear Inverse Problems with Sparsity Constraints. *Journal of Fourier Analysis and Applications*, 14:764--792, 2008. 10.1007/s00041-008-9039-8.
- [56] Yun-Hai Xiao, Qing-Jie Hu, and Zengxin Wei. Modified active set projected spectral gradient method for bound constrained optimization. *Applied Mathematical Modelling*, 35(7):3117 -- 3127, 2011.
- [57] A.A. Palma-Rodriguez, E. Tlelo-Cuautle, S. Rodriguez-Chavez, and S.X. Tan. DDD-based symbolic sensitivity analysis of active filters. In *Devices, Circuits and Systems (ICCDSCS), 2012 8th International Caribbean Conference on*, pages 1 --4, march 2012.