

Optimization of Analog Integrated Circuits Including Variations

by

Adriana Carolina Sanabria Borbón

A Dissertation Submitted to the Program in Electronics Science, Electronics Science Department in partial fulfillment of the requirements for the degree of

Master in Electronic Science

at the

Instituto Nacional de Astrofísica, Óptica y Electrónica August 2014 Tonantzintla, Puebla

Advisor:

Esteban Tlelo Cuautle

©INAOE 2014 All rights reserved The author hereby grants to INAOE permission to reproduce and to distribute copies of this thesis document in whole or in part



The three great essentials to achieve anything worthwhile are: Hard work, Stick-toitiveness, and Common sense.

Thomas A. Edison

Abstract

by Adriana Carolina Sanabria Borbón

Automatic biasing and sizing of analog integrated circuits (ICs) remains an open challenge. This Thesis introduces an automatic technique for sizing analog ICs through combining multi-objective optimization techniques and the g_m/I_D design technique. In this manner, two evolutionary algorithms are described and they are applied as monoand multi-objective algorithms for optimizing analog ICs. They are known as: differential evolution (DE) and non-dominated sorting genetic algorithm (NSGA-II), respectively.

Although current literature summarizes recent sizing techniques for analog ICs, those techniques do not consider the usefulness of exploiting the advantages of biasing techniques, which can enhance DE and NSGA-II algorithms to guarantee DC operating conditions of transistors, and to limit the search space of evolutionary algorithms. That way, this Thesis highlights the advantages of using the g_m/I_D design technique for establishing the sizes, width (W) and length (L) ranges, for each transistor while guaranteeing the appropriate bias levels conditions. The established feasible sizes ranges become the initial search spaces for Ws/Ls when performing automatic IC optimization.

The experiments shown herein, lets us concluding on the appropriateness of applying the g_m/I_D design technique to accelerate the computation time of evolutionary algorithms for optimizing analog ICs. This Thesis discusses the main advantages of this biasing and sizing approach, which are: the search spaces for W/L are feasible values for the given IC technology; the bias conditions of all transistors are guaranteed, and the computing time required by evolutionary algorithms is diminished because the convergence of the algorithms being improved.

Finally, real IC designs not only require accomplishing industrial target specifications, but also they should do it plus guaranteeing robustness, which means the designed IC must support Process, Voltages and Temperature (PVT) variations. This is directly related to yield improvement, i.e. guaranteeing the correct work of a high percentage of fabricated chips. In addition, another strategy to estimate the robustness with respect to parameter variations, like performing sensitivity analysis, is also presented in this Thesis. At the end, the main contribution of this Thesis is the introduction of a multi-objective optimization approach for analog ICs by combining g_m/I_D technique and evolutionary algorithms, and by including PVT variation analysis.

Resumen

by Adriana Carolina Sanabria Borbón

El dimensionamiento automático de circuitos integrados (CIs) analógicos sigue siendo un desafío. Este trabajo muestra una técnica automática para dimensionar CIs analógicos combinando técnicas de optimización y diseño basado en ecuaciones. Primero se describen los algoritmos evolutivos y se explica cómo se usan para resolver este problema de optimización. En este trabajo se han incluido dos tipos de algoritmos: La Evolución Diferencial y el NSGA-II como ejemplos de algoritmos mono y multi-objetivo, respectivamente.

Aunque la literatura resume técnicas actuales de dimensionamiento de CIs analógicos, no considera la utilidad de explotar las ventajas técnicas de polarización antes de iniciar el dimensionamiento automático. De ésta forma, este trabajo muestra la utilidad de la metodología g_m/I_D para calcular los rangos de las dimensiones de cada transistor que garanticen condiciones de polarización dadas. Los rangos factibles se convierten en el espacio de búsqueda inicial de Ws/Ls para la optimización automática. Los experimentos nos permiten concluir que la técnica g_m/I_D es apropiada para reducir el tiempo computacional de los algoritmos evolutivos en la optimización de CIs analógicos. La principales ventajas de esta propuesta son: los espacios de búsqueda son acordes con la tecnología dada, se garantizan las condiciones de polarización y el tiempo de cómputo del algoritmo evolutivo es reducido porque la convergencia del algoritmo se mejora.

Finalmente, el diseño de circuitos reales en entornos industriales requieren no sólo cumplir las especificaciones sino hacerlo a pesar de las variaciones de Proceso, Voltaje y Temperatura. Esto también está relacionado con el aumento del yield, es decir en garantizar que un alto porcentaje de las muestras fabricadas funcionen y cumplan las especificaciones. Por esta razón, una etapa de optimización que tenga en cuenta las esquinas de PVT ha sido incluida en el algoritmo de dimensionamiento desarrollado en este trabajo. Otra estrategia para estimar la robustez a variaciones ha sido incluida en este trabajo la cual está basada en el análisis de sensitividad.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Prof. Esteban Tlelo Cuautle for his guidance in the development of this project. The acknowledgment is extended to his doctoral students Victor Hugo Carbajal and Santiago Rodríguez for their explanations and the code they provided to me to calculate the differential computation.

I want also to thank Dr. Guillermo Espinosa Flores Verdad, Dr. Luis Gerardo de la Fraga and Dr. Carlos Sanchez Lopez for their advices and valuable comments about the thesis.

I am deeply grateful to Prof. Edgar Sanchez and all the people I met during my research stay at Texas A&M University.

My sincere thanks also go to all my professors and classmates at Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) for what I learned from them and the shared experiences.

I also want to thank the researches that share the algorithms implementation in which this work is based.

I would like to thank to my family and friends for all their patience, support and help.

Finally, is my appreciation to the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) and the Consejo Nacional de Ciencia y Tecnología (CONACyT) for its support through the scholarship 492459 and the project funding 131839 - Y.

To my parents Pedro and Nancy and my goddaughter Valentina

Contents

A	bstra	let	ii
R	esum	en	iii
A	cknov	wledgements	iv
C	onter	nts	vi
Li	st of	Figures	viii
Li	st of	Tables	x
1	Intr	roduction	1
-	1.1	Background	3
	1.2	Problem Formulation	4
	1.3	Motivation	5
	1.4	Objectives	5
		1.4.1 General objective	5
		1.4.2 Specific objectives	5
	1.5	Thesis Organization	6
2	Evo	dutionary algorithms	8
	2.1	Evolutionary Algorithms Terminology	8
	2.2	Mono-objective Optimization	9
		2.2.1 Differential Evolution	9
	2.3	Multi-objective Optimization	12
		2.3.1 Non Dominated Sorting Genetic Algorithm NSGA version II \ldots	12
	2.4	EAs in IC's sizing	14
		2.4.1 Fitness function evaluation	15
3	$g_m/$	I_D design technique	18
	3.1	Analog design employing Quadratic model	18
	3.2	g_m/I_D biasing and sizing technique	19
	3.3	ICs sizing combining g_m/I_D and evolutionary algorithms	25

4	Bia	sing and sizing considering PVT Variations	27							
	4.1	1 PVT Variations								
	4.2	2 Corners Analysis								
	4.3	Sensitivity analysis								
		4.3.1 Symbolic Expression computation								
		4.3.2 Example								
		4.3.3 AC sensitivity	33							
		4.3.4 Sensitivity Calculation	34							
5	Exa	mples	37							
	5.1	Optimal sizing for the OTA-Miller amplifier by DE Algorithm	37							
	5.2	Optimal sizing for the OTA-Miller amplifier by NSGA-II Algorithm	40							
	5.3	Optimal sizing for the OTA-Miller amplifier by NSGA-II Algorithm com-								
		paring real vs integer codification								
	5.4	Optimal sizing of a CCII by NSGA-II								
		5.4.1 CCII-based filter working in current-mode	49							
		5.4.2 CCII-based SRCO	50							
	5.5	Optimal sizing for the Recycling folded cascode (RFC) amplifier 51								
	5.6	Optimization employing evolutionary algorithms combined with g_m/I_D . 53								
		5.6.1 Optimal sizing for the OTA Miller amplifier employing DE com-								
		bined with g_m/I_D	53							
		5.6.2 Optimal sizing for the OTA Miller amplifier employing NSGA-II combined with $c_{\rm e}/L$	EE							
	F 77	combined with g_m/I_D	55 56							
	0.1 5.0	Optimization employing evolutionary algorithms under process variations	00 50							
	5.8	Sensitivity Analysis	58							
6	Cor	nclusion	61							

A Publications

Bibliography

List of Figures

1.1	Traditional analog design method $[1]$	2
1.2	Analog design octagon [2]	2
2.1	Evolutionary Algorithms flow chart.	9
2.2	Pareto Front	13
2.3	Analog integrated circuits codification.	14
3.1	g_m/I_D MOSFET characterization setup	20
3.2	Transconductance efficiency	20
3.3	Intrinsic gain	21
3.4	Intrinsic bandwidth	21
3.5	Current density	21
3.6	Intrinsic gain stage circuit.	23
3.7	Look-up table.	24
3.8	Quadratic equation vs g_m/I_D sizing	24
3.9	Combining g_m/I_D technique and NSGA-II	26
4.1	Comparison between under and over design [3]	28
4.2	Flow diagram of the optimization technique considering PVT corners	29
4.3	MOSFET transistor small-signal model	30
4.4	RC example circuit.	32
4.5	Sensitivity of $ H(s) $ w.r.t Cs and G	35
4.6	Sensitivity of $\angle H(s)$ w.r.t Cs and G	36
4.7	RC response under variation	36
5.1	OTA Miller	37
5.2	Comparison of the performance for different values of F and CR for strat-	
	egy 1	38
5.3	Comparison of the performance for different values of F and CR for strategy 6	38
5.4	Comparison of the performance for different values of F and CR for strat-	20
55	egy o	- 39 - 40
5.6	Comparison between DE and NSCA performance	40
5.7	Average constraint violation comparison for DE and NSCA II	41
5.8	Comparison of the solutions of DE and NSCA II algorithms	41
5.0	OTA Millor amplifior	42
5.10	Pareto Front for sizing the OTA Miller amplifier using integer and real	40
0.10	encoding values	44

5.11	Second generation current conveyor (CCII) taken from [4]	47
5.12	Pareto Front for sizing the CCII using integer (black dots) and real (red	
	dots) encoding values.	48
5.13	OTA	48
5.14	Current-mode universal filter based on DOCCIIs taken from [5]	49
5.15	Frequency responses of the universal filter working in current-mode	50
5.16	Time-response of the SRCO computed by SPICE using an optimized CCII.	51
5.17	Recycled OTA.	51
5.18	Recycled OTA solutions	52
5.19	Cost evolution.	54
5.20	Final Population Cost.	55
5.21	Violation constraint average comparison	55
5.22	Best population.	56
5.23	Solutions under variations for corners: TT and SS	56
5.24	Solutions under variations for corners: SF, FS and FF	57
5.25	Violation constraint average comparison.	58
5.26	Sensitivity of $ H(s) $ w.r.t each OTA parameter	59
5.27	Sensitivity of $\angle H(s)$ w.r.t each OTA parameter	59
5.28	Percentage of variation of W(M5)	60

List of Tables

3.1	Comparison of MOSFET performance operating in weak, moderate and	
	strong inversion	:5
5.1	Parameter values	8
5.2	OTA Specifications	9
5.3	Solutions	0
5.4	OTA encoding	4
5.5	OTA results	5
5.6	Last genes values	5
5.7	CCII encoding	7
5.8	CCII results	8
5.9	Last genes values	9
5.10	Circuit element values for the universal filter	0
5.11	OTA	1
5.12	OTA recycling Specs	2
5.13	OTA recycling Results	$\mathbf{b}2$

List of Algorithms

1	Differential Evolution	10
2	Binomial Crossover	11
3	Exponential Crossover	11
4	NSGA-II algorithm [6]	13

Chapter 1

Introduction

Despite the digital integrated circuits (ICs) applications and capabilities grow everyday, analog ICs are still needed when designing mixed-signal systems. For instance, analog ICs are specially required in the input/output stages for signal conditioning circuits, data converters, clock-signals generation and Radio Frequency (RF) applications.

Considering fabrication technologies, the miniaturization processes do not scale in analog ICs, because Moore's law is not as valid for analog as for digital ICs [7]. This means that designing an analog IC with micro-technology, it does not scale down to nanotechnology, as it may be done for digital IC. In addition, compared to digital ICs, analog ones need to accomplish many more target specifications that are more difficult to accomplish than when designing digital ICs. Besides, in all design cases, for analog, digital and/or mixed-signal circuits, their device sizes determine the performance features and their functionality, so that appropriate biasing and sizing techniques are required to improve the design of integrated circuits.

Design automation or manual IC design consists of finding the right topology and its feasible circuit element values that satisfy some target performance specifications. Figure 1.1 sketches traditional IC design flow. Once the topology is selected by the designer (at the transistor level of ICs design) the bias voltages and branch currents are estimated considering some specifications. The next step is the sizing task that is performed using MOSFET equations and models to determine the widths (W) and lengths (L) of the transistors until accomplishing target specifications. After that, circuit simulations are executed to check the circuit behavior and tune it in order to accomplish the desired performance in the different domains, namely: DC, AC and time. However, varying the W and L sizes makes the sizing task to be a difficult one when it is manually executed. In this manner, this task is automated herein by applying evolutionary algorithms and considering process, voltage and temperature (PVT) variations. The biasing conditions of the transistors are guaranteed by applying the g_m/I_D design technique, and also to improve automatic sizing when applying evolutionary algorithms, as described in the following chapters.



FIGURE 1.1: Traditional analog design method [1]

In the electronic design automation (EDA) industry, it is well known that, while the digital ICs design is well supported by sophisticated EDA tools, the same cannot be said for the case of analog EDA tools in several important respects [8]. For example, analog ICs are difficult to design due to the many constraints, and trade-offs that appear particularly for each kind of circuit; further, analog circuit performance features are involved in compromises that make infeasible improving one of them without affecting other or others. Figure 1.2 is an example of the main tradeoffs that an analog IC designer can found in amplifier design [2]. In addition to gain and bandwidth there are some other important specifications for amplifier design like power dissipation, linearity, noise and voltage swings. However, other aspects like input and output impedances are also important because they determines how the circuit interacts with the other stages of the system. The same figure shows how each feature trades at least with another two; in that sense it illustrates the compromises a designer has to balance in analog design.



FIGURE 1.2: Analog design octagon [2]

In consequence, analog design is reluctant to be an automatic process [9]. In this manner, automatic circuit biasing and sizing tasks are currently considered as main research topics

in the EDA business [10]. In this context, several strategies for the automatic biasing and sizing of analog ICs have been summarized in [11]

1.1 Background

Strategies for performing the automatic ICs sizing are classified into two main groups: knowledge based and optimization based. The first one uses a design plan derived from expert knowledge of IC designers [11]. Examples of these techniques are IDAC, OASYS, TAGUS, CAMP, ISAID and BLADES. All these tools are expert systems that employ artificial intelligence to capture the designer knowledge [11]. One of the main advantages is shorter execution time; but the complexity (an independent design plan have to be developed for each type of circuit), long time (the design plan deriving can be many times longer than the manual design process) and high load of deriving the design plan, make these techniques inefficient [12]. Moreover, according to Rocha [11] the obtained results with these techniques are not optimized at all.

Due to the need of developing sizing tools that can provide optimal design solutions and can operate independently of the circuit topology and IC fabrication technology, the next generation of sizing strategies were based on optimization techniques. The word optimization can be defined as the process of finding the best way to use the available resources to obtain a desired result, without violating any constraint imposed by IC technology. [13]. Optimization techniques can also be divided into three main groups, namely: equation-based, simulation-based, and model-based ones [11].

The equation based techniques employs analytical equations that describe the behavior of the circuit under design to evaluate the circuit performance. Examples of these are the tools STAIC, OPASYN, OPTIMAN, ISAAC, DONALD, GPCAD and DARWIN. Nevertheless, these methods are not accurate because the complexity of deriving analytical expressions for all the performance features (for example slew rate) [12]. For this reason, it were proposed another techniques such as FASY, which employ the analytical expressions in order to give a first approach and combines it with simulations to fine tune the solutions [11]. In summary, the main advantage of this method is the short execution time once the design analytical expressions are derived. But in contrast, the approach error in the analytical expressions and the complexity of deriving expressions for all metrics becomes the disadvantage of these techniques.

In simulation-based sizing techniques the circuit's behavior is obtained by numerical SPICE simulations at the transistor level of abstraction and the sizing is performed using optimization techniques. Tools such as DELIGTH.SPICE, FRIDGE, MAELSTROM,

ASTRX/OBLX and ANACONDA belong to this class [11, 12]. Besides, in order to improve the yield, some sizing strategies include layout parasitic extraction. Another authors report the use of evolutionary algorithms to solve the sizing task [14, 15].

Finally, model-based sizing techniques employ macro-models like neural networks, neuralfuzzy, support vector machines (SVM) in order to reduce the execution time due to the less electrical simulator usage in the loop [11].

Other kinds of optimization techniques include: direct search optimization, gradient search optimization, simulated annealing, genetic algorithm and multi-objective optimization [12]. Among them, one can appreciate the usefulness of multi-objective evolutionary algorithms (MOEAs), as the ones applied herein called differential evolution (DE) and Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [16]. Both DE and NSGA-II are linked to a circuit simulator (SPICE, located in the loop) to evaluate the fitness functions of each individual, as already shown in [17–20].

1.2 Problem Formulation

The previous subsection briefly summarized the EDA tools oriented to automate the biasing and sizing of integrated circuits and the main tendencies in automating the analog IC design processes. However, it is still a topic for future research that is aimed to develop new and feasible methodologies for the EDA industry. On the one hand, the increasing fluctuations in IC fabrication processes have introduced an intrinsic uncertainty in circuit performance; therefore, one of the main challenges in IC design is to ensure the fabrication considering parameter and other variations [21]. On the other hand, the main challenges in the development of EDA tools for improving the performance of analog ICs are:

- Computation load: Reduce memory consumption and execution time.
- Level of abstraction: Make optimization at the system level of abstraction.
- Layout: Analog placement, routing and parasitic study.
- To provide feasible sized solutions for the industry in fabrication scale.
- Robustness: Include tolerances, process, volage and temperature variations analysis and design centering.
- Complexity: Reduce the complexity of models or equations.
- Structural and signal path analysis.

• Yield optimization.

The optimization technique developed in this Thesis aims to face some challenges: first, to make automatic sizing of ICs, generating feasible sizes according with the IC fabrication technology; second, to guarantee the sizing being robust to variations, and finally, to reduce the computational load and the execution time.

1.3 Motivation

The motivation of this Thesis is mainly based on the challenges in the analog IC design process, like to find the optimal circuit elements biases and sizes that satisfy all the performance specifications and balances appropriately the trade-offs. However, the IC technology miniaturization includes new challenges related with the reliability and the yield of the circuits. In consequence, nowadays is quite important to consider PVT (Process, Voltages and Temperature) variations in order to provide robustness.

In that way, this Thesis involves the development of an EDA tool for automating the biasing and sizing of analog ICs, employing the direct current (DC) operational point formulae, the sensitivity analysis and the analysis of PVT variations. As a result, and as stressed in [9], analog IC design is reluctant to be an automatic process. For this reason, automatic IC biasing and sizing are currently main research fields in the EDA business, where process and technology variations are the main bottle-necks to mitigate for improving yield [22].

1.4 Objectives

1.4.1 General objective

The main objective of this Thesis is focused on the development of a new EDA tool for the automatic biasing and sizing of analog ICs, based on evolutionary algorithms for performing optimization and DC operating point approach, e.g. g_m/I_D design technique. At the end, the feasible solutions must guarantee the design to be robust to PVT variations.

1.4.2 Specific objectives

• Application of an evolutionary algorithm as an optimization technique to automate the biasing and sizing of analog ICs.

- Employ explicit equations and models of the MOSFET transistor in order to reduce the search space for each sizing parameter and to improve the performance of the optimization algorithms.
- Combine g_m/I_D design technique to evolutionary algorithms for improving biasing and sizing approaches.
- Introduce an automatic approach to guarantee the biases and sizes being robust to PVT variations.

1.5 Thesis Organization

This thesis is organized as follows: Chapter 2 introduces the theory about EAs and the basic terminology related with IC optimization. In this manner, it is outlined the operating principle of two specific EAs for mono-objective and multi-objective optimization. Finally, it is explained how they can be applied to the automatic biasing and sizing of integrated circuits and how the circuits simulator is introduced in the loop.

Chapter 3 is devoted to describe the g_m/I_D design technique and a brief comparison of it with traditional design techniques. After that, it is described how to combine this design methodology with evolutionary techniques. The g_m/I_D technique employs the DC operating point formulation to formulate a proper and reduced search space for each optimization parameter.

On Chapter 4 is presented the challenge of consider process, voltage and temperature variations in analog circuit design and some ways to analyze that. In order to guarantee the solution's robustness to variations it is proposed an additional stage in which optimization is performed considering all the combinations of variations corners. Other way to estimate the variation that a sizing is able to manage is performing sensitivity analysis. For that reason is presented an automatic tool that from circuit netlist extracts the symbolic expressions of the transfer function. Then, sensitivity calculation is explained and a graph-based tool is employed to derive the expressions ans perform the numerical evaluation of the sensitivities with respect of each parameter. Finally the numeric values can be computed in the frequency domain, it is for a range of frequency values. Performing sensitivity analysis is probed being a way to identify the circuit most sensitivity elements, it means that ones with more impact in degrading the whole response.

Through Chapter 5 a set of experiments is presented, including sizing problems of different analog circuits. First, some examples using two evolutionary algorithms are presented as described on Chapter 2, DE and NSGA-II. After that, there are included some examples of combining evolutionary algorithms with the g_m/I_D design technique, and finally a variation analysis is performed with corners optimization and sensitivity analysis examples.

Finally, Chapter 6 describes some conclusions of this Thesis.

Chapter 2

Evolutionary algorithms

As described in [23], evolutionary algorithms are optimization techniques based on the biological evolution and the natural selection of the species theory presented by Darwin. These population based techniques have been developed to solve optimization problems, it means, maximizing or minimizing one or more functions. On these algorithms, an individual is a feasible solution encoded by a chromosome which, in turn, is composed by genes that represent the optimization problem parameters. In the same context, a set of individuals is a population; at the beginning of the algorithm execution an initial population is randomly generated. After that, it evolves iteratively by modifying the chromosomes, through genetic operations such as: crossover, mutation and selection. In each generation only the individuals with better fitness, which is a measure related to some objective functions and constraints, can survive to the next one.

2.1 Evolutionary Algorithms Terminology

According with [13], some important terminology related with EAs is defined:

- Gen: The basic building block in all evolutionary algorithms. Each gen represents an optimization parameter.
- Chromosome: A set of concatenated genes that represent a solution.
- Individual: An aggregate of a chromosome, optimization parameter values x, and objective function (including constraints) value.
- Population: A set of individuals. The solution to a given problem is a set of feasible solutions.

- Fitness function: The measurement of the goodness of a chromosome and it is related with the objectives and constraints. This measurement is employed to compare and classify the individuals that reach the next generation.
- Stop criteria: The criteria that have to be accomplished to stop the algorithm execution.
- Generation: Is the same as one iteration. It also corresponds to each evolution step where a new population is created.
- Operator: Operations between chromosomes in order to generate new ones.

In order to visualize how it works, Figure 2.1 shows the basic flow diagram of any evolutionary algorithm.



FIGURE 2.1: Evolutionary Algorithms flow chart.

In the current literature, there have been reported different kinds of evolutionary algorithms, like differential evolution (DE), non-dominated sorting genetic algorithm (NSGA-II), co-evolutionary algorithms, cultural algorithms [23], and others.

2.2 Mono-objective Optimization

2.2.1 Differential Evolution

Differential evolution is a population based search algorithm proposed by Storn and Price in 1997 [24]. DE is also defined as an optimization technique that belongs to the evolutionary algorithms category [23]. In that sense, DE minimizes multimodal (with not only one solution) cost functions [25]. In this kind of algorithms the cost function uses to be a lineal combination of weighted constraints or objectives.

Like any evolutionary algorithm, DE implementation is based on an iterative procedure that employs some operators, namely: crossover and mutation to modify the population in each generation. In addition, a selection operator is responsible for the choice of the new population's individuals, according to the cost function comparison; in that way only the best adapted individuals reach the next generation.

In DE, the initial population is randomly selected and the gen values are distributed into the search space. Subsequently, the population is modified by the genetic operators in an iterative process until the stop criteria is reached. The stop criteria can be a minimum threshold for the cost function or a maximum number of iterations.

Alg	Algorithm 1 Differential Evolution					
1:	1: procedure $DE(N, g, f_k(x_k))$					
2:	Evaluate the initial population P of random individuals.					
3:	while stopping criterion is not met do					
4:	for each individual in Population do					
5:	Create a candidate c from randomly chosen parent p.					
6:	Evaluate the candidate.					
7:	\mathbf{if} the candidate dominates the parent \mathbf{then}					
8:	The candidate replaces the parent.					
9:	else					
10:	the candidate is discarded					
11:	end if					
12:	end for					
13:	end while					
14:	4: end procedure					

The previous pseudocode rough out the DE procedure, as a function of the population size N, the number of generations g, and the cost function which in turn is dependant of the circuit parameters x. DE working principle is the generation of a mutant chromosome for each individual by adding the weighted difference between two randomly selected population individuals to a third one. It is $y = x_1 + F(x_2 - x_3)$, where F is a scaling factor and x_1, x_2, x_3 are randomly selected from the actual population with the constraint to be different between them. In this evolutionary algorithm, the difference operation allows a gradual exploration of the search space. After mutation the crossover operator takes place. By this operator, parts of the current individual and the chromosome created by mutation are combined in a new one; as a consequence, a trial vector is built according to the crossover probability (CR) [26]. The crossover probability is a control parameter that can take a value from 0 to 1, both included.

After that, a comparison is done between the cost function of both individuals, the trial generated and the current one; only the individual with the best cost value survives to the next iteration [25].

However, this basic principle has been extended to some variants, called strategies. These are conventionally named DE/x/y/z, where DE refers to differential evolution; x is a string that denotes the base individual and can take values such as rand (a randomly selected individual) or best (the individual with the best, usually minimum, cost); yrefers to the quantity of differences implied on the operators, and z refers to the crossover method, which can be binomial or exponential. The most useful differential evolution strategies are described in the references [13, 25, 27, 28].

As mentioned before, the crossover operator can be implemented by either of these options: exponential or binomial. According with [26], the functionality of each one can be described by the following pseudocode:

Algorithm 2 Binomial Crossover

2: $\mathbf{k} \leftarrow \operatorname{irand}(1, \ldots, \mathbf{n}).$ 3: for $j = 1, n$ do 4: if $rand(0, 1) < CR$ or $j = k$ then 5: $z_j \leftarrow y_j$ 6: else 7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	1:	procedure Binomial $crossover((x,y))$
3: for $j = 1, n$ do 4: if $rand(0, 1) < CR$ or $j = k$ then 5: $z_j \leftarrow y_j$ 6: else 7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	2:	$\mathbf{k} \leftarrow \mathrm{irand}(1, \ldots, \mathbf{n}).$
4: if $rand(0,1) < CR$ or $j = k$ then 5: $z_j \leftarrow y_j$ 6: else 7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	3:	for $j = 1, n$ do
5: $z_j \leftarrow y_j$ 6: else 7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	4:	if $rand(0,1) < CR$ or $j = k$ then
6: else 7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	5:	$z_j \leftarrow y_j$
7: $z_j \leftarrow x_j$ 8: end if 9: end for 10: return z 11: end procedure	6:	else
8: end if 9: end for 10: return z 11: end procedure	7:	$z_j \leftarrow x_j$
9:end for10:return z11:end procedure	8:	end if
10: return z11: end procedure	9:	end for
11: end procedure	10:	return z
	11:	end procedure

Algorithm 3 Exponential Crossover

```
1: procedure EXPONENTIAL CROSSOVER((x,y))

2: z \leftarrow x; k \leftarrow irand(1, ..., n); j \leftarrow k; L \leftarrow 0

3: repeat

4: z_j \leftarrow y_j; j \leftarrow (j+1)_n; L \leftarrow L+1

5: until (rand(0,1) > CR) or (L = n)

6: return z

7: end procedure
```

It is also important to check that the new trial generated chromosome parameters are into the search space. For this reason, it is necessary to compare each parameter value with its range boundaries [28].

In a real implementation, DE algorithm control parameters are: the crossover probability CR, the scaling factor F and the population size NP. They are usually fixed as constants

during all the evaluation. However, they can also variate during execution time as already shown in references [29–31].

Among the main advantages of DE algorithm, it is possible to highlight the simplicity of the operation and programming, the good performance to solve multi-modal and multi-variable problems, and the fast convergence [32]. Due to its success, this technique has been extended to other kinds of problems such as the multi-objective ones [33].

DE algorithm has been successfully employed in the design of analog integrated circuits like those that can be reviewed in [34]. In this application field, the cost function can be represented as a combination of the error of several specs. In that sense, if the designed spec has to overload a lower or upper threshold, equations (2.2) or (2.1) respectively can be used to estimate the error.

$$C_A(x) = 1 - \frac{A}{A_{spc}}; A < Aspc$$
(2.1)

$$C_B(x) = \frac{B}{B_{spc}} - 1; B > B_{spc}$$
 (2.2)

The total cost function is a lineal combination of all the specs error, as follows:

$$C_{total} = \alpha_1 C_A + \alpha_2 C_B + \dots \tag{2.3}$$

where α_i are the weight of each error component.

2.3 Multi-objective Optimization

2.3.1 Non Dominated Sorting Genetic Algorithm NSGA version II

The Non Dominated Sorting Genetic Algorithm Version II (NSGA-II) is a non-domination based multi-objective optimization technique (MOO) introduced in 2002 by Deb et al. [16]. The main characteristics of this MOO algorithm are: diversity, convergence and robustness of solutions in the Pareto front (PF) [35]. In these kinds of algorithms there are not only a unique solution, instead there are a set of non dominated solutions that form the Pareto front as showed in Figure 2.2 [24].

NSGA-II approximates the PF of a MOO problem by sorting and ranking all solutions in different Pareto sub-fronts, in order to choose the best solutions to create new offsprings [20]. NSGA-II is summarized by Algorithm 4, in which NSGA-II is described as a function



FIGURE 2.2: Pareto Front

of the population size N, the number of generations g, and the objective functions which in turn is dependent of the circuit parameters x

Alg	gorithm 4 NSGA-II algorithm [6]						
1:	procedure NSGA-II($N, g, f_k(x_k)$) \triangleright N members evolved g generations to solve						
	$f_k(x)$						
2:	Initialize randomly the population P						
3:	Calculate objective values						
4:	Assign rank (level) based on Pareto dominance - sort generated child population						
5:	Binary tournament selection						
6:	for $i = 1$ to g do						
7:	for each Parent and Child in Population do						
8:	Assign Rank (level) based on Pareto - sort						
9:	Generate sets of non-dominated vectors along PF known						
10:	Loop (inside) by adding solutions to the next generation starting from the						
	first front until N individuals found determine crowding distance between points on						
	each front						
11:	end for						
12:	Select points (elitist) on the lower front (with lower rank) and are outside a						
	crowding distance						
13:	Create next generation						
14:	Binary Tournament Selection						
15:	end for						
16:	end procedure						

Then NSGA-II is based on two main procedures: Fast Non-dominated Sort (with $O(MN^2)$ computational complexity) and Crowding Distance Assignment. According to the pseudocode, initially the algorithm builds a population of competing individuals, uses the nondomination level to rank and sort each individual; then applies crossover, mutation and selection operators to create an offspring population; and finally it combines the parents and offspring before partitioning the new combined population into fronts. Additionally, this algorithm includes crowding distance operation in its selection operator and uses it to keep a diverse front by making sure each member stays a crowding distance apart. The crowding distance is a measure of how close an individual is to its

neighbors used to keep the population diversity and helps the algorithm to explore the fitness landscape [6, 36]

2.4 EAs in IC's sizing

The works like [17–20, 37, 38] highlights the usefulness of applying evolutionary algorithms for sizing analog ICs, and also it shows an heuristic technique inspired from fuzzy logic to bound or limit the search space in which the W/L relationships accomplish the desired performance. This is not a trivial task, because, as it was outlined before in the introduction part, in general: analog ICs are difficult to design due to the constraints and trade-offs for each topology and also due to the characteristics of the IC fabrication technology, which do not scales the sizes as IC technology scales down to the nanometer regime. In consequence, as stressed in [9], analog IC design is reluctant to be an automatic process. For this reason, automatic IC sizing is currently one of the main research fields in the Electronic Design Automation business, where process and technology variations are the main bottle-necks to mitigate for improving yield [22].

As a field in research, some attempts of applications have been reported in literature. For this application, the optimization problem is the analog integrated circuit sizing. In this context, the optimization parameters are transistor sizes W and L; biasing conditions and element values like C and R. Additionally, the fitness function is a combination of constraints and objectives, it means, circuit performance specifications.

Figure 2.3 shows a simple example of the circuit codification employed to build each chromosome. In the current mirror, each W and L is assumed as a gene (different L in a current mirror is not a practical choice, but in this example is used to explain how the codification is performed), and the concatenation of all genes conform the whole chromosome.



FIGURE 2.3: Analog integrated circuits codification.

For each parameter, the circuit designer fixes a search space, it means a range in which parameters can get a value. This range must be a feasible set of values according with the IC fabrication technology. It is also assumed the same for the same kind of parameters. For example, all MOSFET channel widths Ws have usually the same search space, and the same is assumed for all channel lengths Ls.

2.4.1 Fitness function evaluation

Due to the objectives and constraints in the case of analog circuit optimization corresponds to circuit performance specifications, one of the most common forms to measure these involves a circuit simulator. In consequence, the evaluation function is updated by linking the circuit simulator SPICE. In the algorithm implementation (in C language in this case) has been programmed a function to perform the evaluation. This function writes the parameter values in the simulation field, then it performs the simulation execution by a system call, and finally it reads the objectives and constraints data from the output files.

For the evaluation stage in this implementation three data files have been employed: the model that includes the BSIM technology parameters of the NMOS and PMOS transistors; the kernel file that contains the netlist of the IC to be sized; and the testbench file.

The kernel file specifies the netlist of the circuit under optimization alike SPICE (simulation program with IC emphasis). The circuit under design is declared as a subcircuit so the biasing, input and output nodes are declared as inputs and output of the subcircuit. All MOSFETs are parameterized by their design variables Ws and Ls using the command *.param*.

Traditional real encoding is done by describing the MOSFET sizes using real numbers, as for the following SPICE file:

```
...
.param p1=900u
.param q1=2u
.param p2=147u
.param q2=.5u
...
MP1 drain_node gate_node source_node bulk_node PMODEL W='p1' L='q1'
MN1 drain_node gate_node source_node bulk_node NMODEL W='p2' L='q2'
MP2 drain_node gate_node source_node bulk_node PMODEL W='p1' L='q2'
...
```

In this case, the optimized sizes Ws and Ls, associated to the real numbers p_i and q_i , should be rounded to be multiples of the IC technology. For example, for an IC

technology of 0.18μ m, neither the p_i s nor the q_i s given above are multiples of 0.9μ m (lambda of the IC technology), so that a post-processing step is needed to round those real values to be multiples of 0.9μ m.

In order to avoid a rounding-off stage, it is used an integer encoding for the Ws and Ls sizes of the MOSFETs, it is performed by assigning integers to p and q, and those integers will be multiplied by the IC technology by using the command *.option scale* within SPICE. In this manner, using an IC technology of 90nm, the proposed integer encoding updates the SPICE file described above, by the following one:

```
...
.option scale=90nm
.param p1=100
.param q1=2
.param p2=140
.param q2=50
...
MP1 drain_node gate
```

```
MP1 drain_node gate_node source_node bulk_node PMODEL W='p1' L='q1'
MN1 drain_node gate_node source_node bulk_node NMODEL W='p2' L='q2'
MP2 drain_node gate_node source_node bulk_node PMODEL W='p1' L='q2'
...
```

In this case, the Ws and Ls sizes of the MOSFETs are scaled by 90nm (lambda). For example, for MP1 .. W = 100 * 90nm (9um) L = 2 * 90nm (0.18um), and so on.

The main advantage of applying integer numbers instead of real numbers in the encoding, is the elimination of a post-processing step for rounding the W and L sizes to multiples of the IC technology, as recently discussed in [4]. Therefore, by using integer encoding in the evolutionary operators the computation time and the memory usage are also decreased.

By applying evolutionary algorithms, as was shown above, the W and L sizes for each MOSFET are parameterized and then encoded by a gen that includes parameters p and q. The chromosome of an analog IC consists of the whole group of genes (MOSFETs), and one chromosome is associated to one individual in the population N.

In other part, the test bench is developed in order to measure the performance features that corresponds to the objectives and constraints of the analog IC. This file includes the subcircuit (kernel) instances, the commands to make the desired analysis (.AC, .TRANS, .DC) and the .MEASURE command to extract the specific values of the constraints and objectives in SPICE format described as follows:

. . .

.OP .AC dec 100 1 100G .MEASURE AC UnitGainF when vdb(vo1)=0 .MEASURE AC DCgain MAX Vdb(vo1) .MEASURE AC PhaseM MIN vp(vo1) from=1 to=UnitGainF .MEASURE AC BW when Vdb(vo1)=' DCgain - 3' ...

Additionally, in order to perform a single spice call for generation, the set of parameters (chromosome) for each individual is included in one line of the test bench file starting with *.param* command, and the lines are separated by *.ALTER* command, as shown below:

```
...
.param p1=.. p2=.. p3=.. **Individual 1
.ALTER
.param p1=.. p2=.. p3=.. **Individual 2
.ALTER
.param p1=.. p2=.. p3=.. **Individual 3
...
```

In that way, the test bench file has as .param lines as individuals. Then, for a population of N individuals, N SPICE simulations are executed in one system call for each generation. However, after the execution a single output file is created for each individual. In that way, each output SPICE file has the label associated to each chromosome to extract the performances (objective function) values, and keeping a pointer to the design variables W and L for each individual.

Chapter 3

g_m/I_D design technique

In traditional IC design tasks, first, a circuit topology is selected, i.e. an architecture composed by the circuit elements and their connections among them is chosen. Then, explicit MOSFET equations are usually employed by analog IC designers to perform biasing and sizing tasks. In this context, different MOSFET models have been reported, but in recent times these have became more complex. This tendence is due to the technology miniaturization; devices properties can not be scaled, so its behavior can not be described and predicted accurately by the older models. In fact, new models has more parameters than older ones, in order to consider secondary effects, like short length effects, that are relevant only in new fabrication sizes.

In other part, automatic sizing of analog ICs requires an initial search space, it is a range in which a variable can take a feasible value. Usually, the search space is initialized by the designer as the same for all variables of the same type (for instance, Ws of all transistors). However, there are not an specific criteria the designer can use to estimate a proper range. A wider range can make the computation time longer because the algorithm need to try more possible values. However, a narrow range can cause non convergency because the feasible (or optimal) solutions can be out of that range.

For that reason, in order to approximate a feasible search space for each optimization variable, some MOSFET equations and models are employed.

3.1 Analog design employing Quadratic model

Quadratic model for a MOSFET is composed by independent equations that describe the behavior for each transistor region: cutoff, linear and saturation region. For example, for a MOSFET operating in saturation region, it means $V_{GS} > V_T$ and $V_{DS} > V_{DSsat}$ and without considering the velocity saturation effects (these effects are detailed explained in [39]), the drain current is described by

$$I_D = \frac{1}{2} \frac{\mu C'_{ox}}{n} \left(\frac{W}{L}\right) (V_{GS} - V_T)^2$$
(3.1)

where the mobility μ , the oxide deep C'_{ox} , the slope factor n and threshold voltage V_T are technological parameters. Also in the equations are involved biasing conditions like V_{GS} and I_D .

Manipulating the previous equation, the MOSFET transconductance can be derived as

$$g_m = \frac{\delta I_D}{\delta V_{GS}} = \frac{\mu C'_{ox}}{n} \left(\frac{W}{L}\right) (V_{GS} - V_T) = \sqrt{2I_D \left(\frac{\mu C'_{ox}}{n}\right) \left(\frac{W}{L}\right)} = \frac{2I_D}{V_{GS} - V_T}$$
(3.2)

Finally, the shape factor can be derived as,

$$\frac{W}{L} = \frac{ng_m^2}{2\mu C'_{ox}} \frac{1}{I_D}$$
(3.3)

However, some transistor models ignore short channel effects such as mobility degradation, Drain-induced barrier lowering (D.I.B.L.) and gate length modulation. On the other hand, there exist more complex models that take into account all these effects that have too much parameters, but they make the manual calculation infeasible. Additionally, advantages of weak and moderate inversion are become this operation regions more exploited for different low power applications.

In this design technique, symbolic expressions for specifications are derived as functions of component parameters. For instance, gain expression in amplifiers is usually associated with the input transistor's transconductance and the impedance of the output stage. After that, for a predefined operating region, model equations and biasing conditions are used to stablish the proper voltage and current levels and element sizes.

3.2 g_m/I_D biasing and sizing technique

The g_m/I_D design technique includes all the operation regions of the MOSFET transistor. It was proposed for first time by Silveira [40].

According to (3.1) and (3.2), both terms, transconductance (g_m) and drain current (I_D) are proportional to the transistor width (W) and length (L), so that the ratio between

 g_m and I_D does not depend on the transistor sizes [41], but only on the DC biasing. From the same equations the transconductance efficiency can be derived as:

$$\frac{g_m}{I_D} = \sqrt{\frac{2}{I_D} \left(\frac{\mu C'_{ox}}{n}\right) \left(\frac{W}{L}\right)} = \frac{2}{V_{GS} - V_T}$$
(3.4)

Additionally, g_m/I_D shows a relationship between the small and large signal [41], and several performance features can be described in terms of g_m/I_D .

To characterize a fabrication IC technology, a simulation of the IC is performed to measure transistor parameters such as threshold voltage (V_{th}) , transconductance efficiency g_m/I_D , intrinsic transistor gain (g_m/g_o) , current density (I_D/W) , and transit frequency (f_T) for a set of channel length values when gate to source voltage V_{gs} is changing. The characterization data is saved in look-up tables and can be visualized in Figures 3.2 to 3.5.



(a) NMOS characterization setup (b) PMOS characterization setup

FIGURE 3.1: g_m/I_D MOSFET characterization setup



FIGURE 3.2: Transconductance efficiency

Based on the fact that transconductance and drain currents are proportional to the channel width, one can define another useful expressions in order to obtain specific values



FIGURE 3.3: Intrinsic gain



FIGURE 3.4: Intrinsic bandwidth



FIGURE 3.5: Current density

employing data extracted from the characterization measurements of the MOSFET. In this manner, from the specifications are extracted desirable values for g_m/I_D and g_m for a determined L. For instance, the desired MOSFET transconductance can be extracted from f_T specification and capacitance value as follows:

$$g_m = w_t * C = 2 * f_t * C \tag{3.5}$$

Then, for a known g_m , and considering the denominator values from the characterization saved in LUT, the drain current can be derived using

$$I_D = \frac{g_m}{\left(\frac{g_m}{I_D}\right)^*} \tag{3.6}$$

where the denominator corresponds to the efficiency of transconductance saved on the LUT.

After that, the specifications can be completely mapped in a current density value I_D/W , so that the W value can be derived from

$$W_x = \frac{I_D}{\left(\frac{I_D}{W}\right)^*} \tag{3.7}$$

where the denominator corresponds to the current density stored on the LUT.

In general, the method proceeds according to the steps described below [41]:

- 1. Set-up look-up tables (LUTs) making use of Spice or BSIM.
- 2. Choose primary variables.
- 3. Estimate gate lengths in accordance with the desired gain and transit frequencies. Short channel: Small capacitances, high f_t (high speed), less layout area Long channel: high intrinsic gain, good matching, reduction of flicker noise
- 4. Evaluate parasitic capacitances.
- 5. Evaluate currents and widths taking advantage of the target specifications. Since closed form solutions cannot be found generally in complex circuits, some assumptions may be required wherever necessary.
- 6. Re-iterate drain and width evaluations to get rid of the assumptions.
- 7. Check the result by running a circuit simulator like Spice.

An IC design test can be developed in order to illustrate the accuracy of an IC design using different techniques. That way, in order to compare the effectiveness of g_m/I_D overt he traditional quadratic design model, an example is given. Let us consider the sizing of a common source amplifier, also known as intrinsic gain stage shown in Figure 3.6, according with these Specs: Technology of 0.18 μ m IBM, unity-gain frequency of 100 MHz, DC gain of 4V/V or 12.04 dB, capacitive load= 1pF, and voltage supply Vdd=1.8V.



FIGURE 3.6: Intrinsic gain stage circuit.

The voltage gain of the intrinsic gain stage is approximated to

$$A_v = -g_m R \tag{3.8}$$

The dominant pole frequency can be approximated by the expression

$$\omega_p = \frac{-1}{RC_L} \tag{3.9}$$

Then, the unity-gain frequency can be obtained as

$$\omega_u = A_v \omega_p = \frac{g_m}{C_L} \to g_m = \omega_u C_L = 2\pi f_u C_L \tag{3.10}$$

According with the specifications, the transconductance is

$$g_m = 2\pi (100MHz)(1pF) = 628.31e^{-6}$$
(3.11)

The resistor value can be derived as

$$A_v = -g_m R \to R = \frac{A_v}{g_m} = 7639 \tag{3.12}$$

and the drain current value is

$$I_D = \frac{0.9}{7639} = 117.8\mu A \tag{3.13}$$

So, employing the quadratic model equation for MOSFET device operating in saturation region

$$\frac{W}{L} = \frac{n(g_m)^2}{2(\mu C_{ox}I_D)} = 15.43 \tag{3.14}$$

Finally, assuming n=1.2, and the minimum channel length L=180nm, W=2.77 μm

on the other hand, employing the g_m/I_D technique, the efficiency of transconductance value can be derived as

$$\frac{g_m}{I_D} = \frac{628.31e^{-6}}{117.8e^{-6}} = 5.33\tag{3.15}$$

The $\frac{g_m}{I_D}$ value is associated to some specific characteristic in the LUT, as shown in Figure 3.7.

Vgs[X]	Leff[Y]	Vth[Y]	Vov[Y]	gm/ld[Y]	gm/go[Y]	Id/W[Y]	ft[Y]
0.764	1.45e-07	0.503174	0.260826	5.43152	16.8008	72.3989	48,032,200,000
0.765	1.45e-07	0.503141	0.261859	5.4075	16.7828	72.8158	48,095,600,000
0.766	1.45e-07	0.503108	0.262892	5.38366	16.7649	73.2333	48,158,500,000
0.767	1.45e-07	0.503075	0.263925	5.35998	16.7471	73.6513	48,220,900,000
0.768	1.45e-07	0.503042	0.264958	5.33646	16.7293	74.07	48,282,700,000
0.769	1.45e-07	0.503009	0.265991	5.31311	16.7117	74.4891	48,343,900,000
0.77	1.45e-07	0.502976	0.267024	5.28993	16.6941	74.9089	48,404,600,000
0.771	1.45e-07	0.502943	0.268057	5.2669	16.6766	75.3291	48,464,800,000
0.772	1.45e-07	0.50291	0.26909	5.24403	16.6592	75.7499	48,524,400,000

FIGURE 3.7: Look-up table.

Assuming again the minimum channel length,

$$W = \frac{I_D}{\left(\frac{I_D}{W}\right)^*} = \frac{117.8\mu A}{74.07} = 1.59\mu m \tag{3.16}$$

The sizing is validated with the simulation in frequency domain shown in Figure 3.8. From this figure it is clear that employing g_m/I_D technique for sizing is more accurate than by using the quadratic model.



FIGURE 3.8: Quadratic equation vs g_m/I_D sizing.
Spec.	Weak I.	Moderate I.	Strong I.
Shape factor	large	medium	small
Capacitance	large	medium	small
Bandwidth	low	moderate	high
DC leakage	high	moderate	low
$\frac{g_m}{I_D}$	high	moderate	low
gain	high	moderate	low
$V_{EFF}, V_{DS,sat}$	low	moderate	high

 TABLE 3.1: Comparison of MOSFET performance operating in weak, moderate and strong inversion.

 $*V_{EFF} = V_{GS} - V_{TH}$

3.3 ICs sizing combining g_m/I_D and evolutionary algorithms

Although [11] summarizes current sizing techniques for analog ICs, those works and also [9, 42], do not consider the usefulness of exploiting biasing techniques advantages [43], before starting the automatic sizing. In this manner, this Thesis shows the usefulness of the g_m/I_D design technique [41], for computing sizing ranges for each transistor that guarantees some bias levels conditions, a similar work as the one already done in [42], but now applying the g_m/I_D design technique has the advantage that it guarantees biasing conditions. The feasible ranges become the initial search spaces for Ws/Ls for the multiobjective optimization algorithms DE and NSGA-II, described previously, in Chapter 2.

The proposed approach is depicted in Fig. 3.9, shows the procedure mean the g_m/I_D and evolutionary algorithms (NSGA-II in this case) are combined. The first step consists on IC technology characterization, in order build the LUT with the transistor parameters mean an voltage sweep and for different transistor lengths as described before. Then, an operating region for MOSFETS and biasing nodal voltages conditions are assumed in order to figure a V_{GS} range of each transistor. The operation region is a criteria assumed by the designer; each of the operation regions: weak, moderate and strong inversion offer different advantages in terms of performance as shown in Table 3.1. In order to estimate the biasing conditions is necessary to take into account the biasing voltages, and the input and output swing.

For instance, according with [44] the operating region of MOSFET devices of an amplifier can be chosen as: a large input pair biased in weak/moderate inversion to maximize bandwidth and minimize noise and offset, and current mirror devices with long channels biased in strong inversion to improve mirroring accuracy and output impedance. Some specifications can also be included in order to determine feasible ranges for parameters such as I_D , g_m , and parasitic capacitances. After that, using LUT's data and g_m/I_D technique equations, all V_{GS} ranges are mapped into W ranges.

At this point, there are a delimitated search space for each optimization parameter that accomplish the biasing conditions and that is feasible for the IC technology. This search spaces are employed like initial conditions for the optimization algorithm. It means, each variable has now its own range which is different to the range of other variables. This range defines the maximum and minimum values a variable can take at the initialization part and after each operator application.



FIGURE 3.9: Combining g_m/I_D technique and NSGA-II.

In the optimization stage, the inputs of the algorithm are the circuit netlist (as described in the subsection 2.4.1); the objectives and/or constraints, which are related desirable specifications of the circuit; and the range of each variable. Meanwhile the outputs are different sizes combinations where all of them accomplish the target specifications. Also, the evaluation step requires performing circuit simulations, which at the same time includes the modification of the Spice netlist, the performance of circuit simulation and the reading of specs measurements from output files.

Chapter 4

Biasing and sizing considering PVT Variations

4.1 PVT Variations

The analog IC design industry has facing several important challenges related to some issues. For example, how to achieve the best specifications; second, how to reduce silicon area and, how to reduce power consumption, mainly; on the other hand, other issues are how to reduce the time to market and guarantee a high yield. According to the last requirement, besides an IC design accomplishes target specifications, it must be also robust to parameter and process variations. For instance, a measure to estimate robustness is reflected in the yield, which is defined as the ratio between the amount of fabricated ICs that effectively work and accomplish all the specifications with respect to the total fabricated ICs.

Yield decreases because the IC design does not support variations. During the fabrication process and during the operating time of the IC, an integrated circuit modifies its response with respect to the target specs because a designer neither can control nor can predict parameter and process variations.

Variations can be classified according to their causes in three main groups: due to voltage, temperature or process. Another variations sources like aging have been introduced in the literature but they are not case of study in this research.

With respect to process variations, electrical devices variations depend on global variation (between die, wafers, and lots), local variation (i.e. mismatch, uncorrelated atomistic variation of each device instance), and layout dependent effects [45]. So, considering

variation issues is the best way to improve the yield, but sometime it also can lead us to perform an over-design.



FIGURE 4.1: Comparison between under and over design [3]

Design considering variations is not a trivial task. As is depicted in the Figure 4.1, if variations are not properly handled they can occur two scenarios [3]. In the first one, due to a excessive safety a compromise between performance/power and area, this case is called over-design; in the second one, due a relaxed or minimal consideration of variations there is no sacrifice in power and area but the circuit might not met the specifications under variations, which can derives in a poor yield, it it called under design [3]. Neither of the described scenarios is desirable because all: power, area and yield in the industry must be optimized, or at least well balanced.

Process variation is usually modeled in two ways:

- Process corners
- Process Monte Carlo

4.2 Corners Analysis

IC designers need to estimate the variation in a circuit performance due to process variations, for this reason, ICs fabricant provides data about the corners of MOSFET devices in their simulation models based on measured data. Usually, the variations have been defined by corners as: T, S, and F, for typical, slow, and fast, respectively and for MOSFET type N and P. So the following combinations are possible: FF, FS, TT, SF and SS. Additionally, for voltage and temperature variations there are defined two corners and the nominal value. A common used ranges for temperature are -20, 60 and 100 celsius degrees meanwhile for voltage it is $\pm 10\%$ the nominal value.

The variations simulations should consider all the combinations for: five process corners, three voltage corners and three temperature corners. So, the total amount of corner combinations is the product of all the possible values: five by the square of three, it is 45; then, in order to evaluate each individual fitness is necessary to perform 45 simulations for individual instead of one.

Because of the corner analysis, computational time increases significantly so, it is necessary to figure out the way to find solutions with the least number of simulations. One way to do it is to simulate first the typical case and the nominal values, if the constraint violation is equal to zero, it means if an individual accomplish all constraints at nominal values, so it is feasible to evaluate the corners, otherwise it is not practical.

For this reason, the final implementation of the evolutionary algorithm showed in the Figure 4.2 has two search spaces downsizing stages. The first one, applying g_m/I_D technique and the second one after running the algorithm and all the population individuals reach all the constraints. The first case was already explained in the Section 3.3 and the second one is implemented running the algorithm and until to accomplish the stop criteria, in this case when the average constraint violation of all the population is equal to zero, it is all the individuals of the population accomplish the constraints.



FIGURE 4.2: Flow diagram of the optimization technique considering PVT corners

4.3 Sensitivity analysis

The frequency response of a linear circuit can be described by a transfer function that can be further used to perform a sensitivity analysis, in order to identify sensitive circuit elements. That way, one can know that when a circuit element behavior changes, it modifies the response of the whole circuit. In other words, sensitivity is a measure of the variation of a circuit as a whole, due to the variation of a parameter or circuit element.

In general, the majority of sensitivity analysis techniques needs as input the transfer function that can be obtained by performing linear algebraic or graph operations. On the other hand, to get a better insight on the behavior of a circuit, performing symbolic analysis is quite useful for deriving analytical expressions of analog integrated circuits (ICs), and thus to improve their design as already shown in [46]. Deriving transfer functions is a good starting point for characterizing the dominant behavior of an analog IC [47], which can help to select the best conditions looking for an optimal design [48]. In this manner, several symbolic techniques have been developed to derive analytical expressions or to transform certain characteristics of linear networks, see for instance [49– 51]. In particular, the work in reference [51], shows an improved graph-based symbolic technique for performing sensitivity analysis of analog ICs.

4.3.1 Symbolic Expression computation

To perform symbolic analysis, there are several techniques ranging from using different kinds of circuit-element models to distinct types of circuit descriptions [46]. In particular, circuit with differential characteristics are quite simple to analyze by using nullors and pathological circuit elements, as shown in [52]. In this work, analog ICs are modeled using traditional voltage controlled models. In that way, the behavior of the transistor can be modeled as a linearization of the large signal model, resulting in the small signal equivalent circuit of the figure 4.3 discussed in [53]. In this model some parameters such as the one related with the bulk terminal is neglected to facilitate the application of symbolic sensitivity analysis.Consider more parameters can lead in a more accurate approximation of the MOSFET behavior, but it is necessary to be aware because the increase of the circuit complexity too.



FIGURE 4.3: MOSFET transistor small-signal model.

A symbolic analyzer was developed in this work. It was programmed in C language and has as input the netlist of a circuit like SPICE. First, the program identify the kind of element from the netlist, and extracts and all the current and voltage sources. These are separated between biasing and signal sources; then, biasing sources are disconnected, it means the voltage sources are replaced by a short circuit and the current sources by an open circuit. Signal sources are setted in the sources and variables vector as corresponds. Then, the quantity of elements is calculated, and the program extract the nodes of all elements forming the variables vector. Also name and value information is extracted for each element and stored in a data structure. Afterwards, the MNA stamps are employed to introduce each element in the admittance matrix. According to the nodes information, the modified nodal analysis expression is formed as:

$$Yx = S \tag{4.1}$$

where Y is the admittance matrix, x the variables vector and S the sources vector.

In order to get the transfer function the Cramer's rule is employed. The user can choose the output node and then, the program extracts both matrixes: the admittance matrix and the admittance matrix modified with the sources vector.

Finally, the admittance matrix is presented in a suitable form to perform the sensitivity analysis. Assuming R_i and C_i as the row and column indexes, respectively; *s* represents the sign of the element which take values of 1 or -1 for positive and negative respectively; and the name and value are also included, the admittance file has the form:

> $R_1 C_1$ s name1 value1 $R_2 C_2$ s name2 value2

4.3.2 Example

The circuit shown in Figure 4.4 is employed to explain the developed software. This simple circuit was chosen only for obtaining all the symbolic expressions in an easy way, but in fact, the developed software can deal with more complex circuits.

The transfer function of this circuit can be easily derived as:

$$H(s) = \frac{1}{1 + rcs} \tag{4.2}$$

Replacing R=1/, the transfer function becomes,



FIGURE 4.4: RC example circuit.

$$H(s) = \frac{g}{g+cs} \tag{4.3}$$

So now, the procedure used by the algorithm will be explained. First, the circuit should be described in a Spice form, for the current example, it is:

V n1 0 AC 1 R n1 no 1e3 C no 0 10e-12

Then, the modified nodal analysis is summarized in the following equation:

$$\begin{pmatrix} g & -g & 1 \\ -g & g+sC & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} n1 \\ no \\ iV \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ V \end{pmatrix}$$

With the objective of solve the system for the n_o node employing Cramer's Rule, that is:

$$x_i = \frac{\det(Y_i)}{\det(Y)} \tag{4.4}$$

where x_i is a system variable, and Y_i results changing the i_{th} column by the sources vector. So, in the algorithm all resistor elements are changed to conductances and the admittance matrix description obtained by the software becomes:

0 0 1 g 1.00000e-03 0 1 -1 g 1.00000e-03 0 2 1 1 1.00000e+00 1 0 -1 g 1.00000e-03 1 1 1 g 1.00000e-03 1 1 1 s*c 6.283200e-10

$2\ 0\ 1\ 1\ 1.00000e{+}00$

The first column is the matrix row index, the second one is the matrix column index (starting on zero). The third value represents the sign of the element Additionally, the Y_1 matrix is defined as the admittance matrix modified with the source vector, as

	$0 \ 0 \ 1 \ g \ 1.000000e-03$
$\begin{pmatrix} g & 0 & 1 \end{pmatrix}$	$0\ 2\ 1\ 1\ 1.00000\mathrm{e}{+00}$
-g 0 0	1 0 -1 g 1.00000e-03
$\left(\begin{array}{ccc} 1 & V & 0 \end{array}\right)$	$2\ 0\ 1\ 1\ 1.000000\mathrm{e}{+00}$
	$2\ 1\ 1\ V\ 1.00000e{+}00$

4.3.3 AC sensitivity

The sensitivity can be calculated from the transfer function H(s) = N(s)/D(s) using the expressions already discussed in [51], and shown below. According to [9], the sensitivity is described by the first order derivates of the objectives, in this case the performance features, with respect to parameters of the circuit elements.

The normalized sensitivity can be expressed as:

$$Sens(H(s), W) = \frac{W}{H(s)} \frac{\partial H(s)}{\partial (W)}$$
(4.5)

where W is the sensitivity parameter. This expression can be decomposed by dealing with N(s) and D(s) directly, instead of H(s), leading to

$$\operatorname{Sens}(H(s), W) = \left(\frac{WD(s)}{N(s)}\right) \left(\frac{\frac{\partial N(s)}{\partial (W)}D(s) - N(s)\frac{\partial D(s)}{\delta (W)}}{D^2(s)}\right)$$
(4.6)

a simplification leads to the reduced formulae,

$$\operatorname{Sens}(H(s), W) = W\left(\frac{1}{N(s)}\frac{\partial N(s)}{\partial W} - \frac{1}{D(s)}\frac{\partial D(s)}{\partial (W)}\right)$$
(4.7)

Henceforth, employing this last expression in (4.7), it is easy to compute symbolically the sensitivity of the characteristics with respect to each circuit element. At this end, it can be replaced the symbolic terms (circuit elements) by their numerical values computed by Spice. It is worth mentioning, that this research employed the sensitivity identities defined on [54]. For a due transfer function H(s), and a its parameter p are valid the next identities:

1. Let G=1/RSens(H(s), R) = -Sens(H(s), G)(4.8)

$$Sens(|H(s)|, p) = Re\{Sens(H(s), p)\}$$
(4.9)

3.

2.

$$\operatorname{Sens}(\angle H(s), p) = \frac{1}{\angle H(s)} Im\{\operatorname{Sens}(H(s), p)\}$$
(4.10)

4.

$$Sens(H(s), p) = Sens(H(s), ps)$$
(4.11)

Employing the equation (4.7) and the identities, for the same circuit of Figure 4.4, the sensitivity expressions can be easily derived as:

$$Sens(H(s),g) = \frac{Cs}{g+Cs}$$
(4.12)

$$Sens(H(s), Cs) = \frac{-Cs}{g+Cs}$$
(4.13)

According to the first sensitivity identity, the sensitivity w.r.t. R is equal to the sensitivity w.r.t. G by a factor of -1. Also, according with the fourth identity, the sensitivity is the same for C and Cs. So, in consequence, the sensitivity expression is exactly the same for both w.r.t. Cs and R.

Sensitivity Calculation 4.3.4

The software developed employs the derivate matrices Y and Y_i and the graphs based method for the sensitivity evaluation developed in [51].

$$\det(Y) = -(g + sC) \tag{4.14}$$

$$\det(Y_1) = (-gV) \tag{4.15}$$

So, the transfer function is due to:

$$H(s) = \frac{Vo}{V1} = \frac{g}{g+sC} = \frac{N(s)}{D(s)}$$
(4.16)

$$\frac{\partial \det(Y)}{\partial g} = -1 \tag{4.17}$$

$$\frac{\partial \det(Y)}{\partial sC} = -1 \tag{4.18}$$

$$\frac{\partial \det(Y_1)}{\partial g} = -1 \tag{4.19}$$

$$\frac{\partial \det(Y_1)}{\partial sC} = 0 \tag{4.20}$$

According with the equation (4.7), the sensitivity of the transfer function with respect of each parameter can be calculated as

Sens
$$(H(s), g) = g\left(\frac{1}{g}(1) - \frac{1}{g+sC}(1)\right) = \frac{sC}{g+sC}$$
 (4.21)

$$Sens(H(s), sC) = sC(\frac{1}{g}(0) - \frac{1}{g + sC}(1)) = \frac{-sC}{g + sC}$$
(4.22)

For example, values: $R = 1k\Omega$ and C = 10pF, the sensitivity as function on the frequency was calculated. The Sensitivity with respect to the magnitude and the phase of the transfer function are showed in Figures 4.5 and 4.6, respectively:



FIGURE 4.5: Sensitivity of |H(s)| w.r.t Cs and G.

In order to probe the sensitivity results, the RC circuit was simulated for variation of R and C of $\pm 20\%$ each. From the results in Figure 4.7, it can be probed that changing R and C values in the same percentage have the same effect in the circuit response.



FIGURE 4.6: Sensitivity of $\angle H(s)$ w.r.t Cs and G.



FIGURE 4.7: RC response under variation

Chapter 5

Examples

5.1 Optimal sizing for the OTA-Miller amplifier by DE Algorithm

As a mono-objective optimization example, this approach is being employed to size a two-stages operational amplifier with compensation network, as illustrated in Figure 5.1.



FIGURE 5.1: OTA Miller.

The DE kernel employed in this example was developed by Dr. Rainer Storn and Kenneth Price at Berkeley university. One important issue is to determine a proper value for the DE algorithm parameters F and CR, because they are problem dependable. For that reason, the different strategies presented in [25] were tested; Figures 5.2 to 5.4 show the response of the strategies with better convergency. The strategy that showed the better convergency is described by DE/best/1/exp:

$$V_{G+1} = x_{best} + F(x_1 - x_2); (5.1)$$

Parameter	Value
Technology	TSMC180nm
$V_{DD} = -V_{SS}$	0.9V
CL	3pF
Ibias	$100\mu A$

TABLE 5.1: Parameter values.

In addition, to stablish the best combination of values for F and CR for this kind of problem some simulations were executed for all the combinations setting F = 0.5, 0.7 and 0.9, while CR = 0.3, 0.6 and 0.9. Figure 5.2 exhibit the comparison of the convergence of the same strategy for the different combination of parameters.



FIGURE 5.2: Comparison of the performance for different values of F and CR for strategy 1.



FIGURE 5.3: Comparison of the performance for different values of F and CR for strategy 6.

The conditions employed in these simulations are described in Table 5.1, and the specifications are presented in Table 5.2.



FIGURE 5.4: Comparison of the performance for different values of F and CR for strategy 8.

TABLE 5.2: OTA Specifications.

Parameter	Value
DC Gain	60dB
BW	100KHz
Phase Margin	60
Power	1mW
Vos	$100\mu V$
SR(+)	$20V/\mu s$
SR(-)	$20V/\mu s$

All these specifications are formulated like constraints. According to [55], the constraints for the upper and lower threshold can be formulated by 5.2 and 5.3, respectively.

$$g_o(x) = 1 - \frac{c_i}{c_{max}} \tag{5.2}$$

$$g_o(x) = \frac{c_i}{c_{min}} - 1 \tag{5.3}$$

On the other hand, each constraint value is compared with the specification umbral. The cost of each specification (calculated by equations (5.2) and (5.3)) is only added to the global cost function only if the specification is not reached.

The algorithm was executed for a 100 individuals population and with a stop criteria of maximum number of iterations 150. The average cost function was calculated for each generation (iteration) and plotted in Figure 5.5.

From the previous figure can be verified the cost function is always descendent and is has the tendency to reach a minimum value.



FIGURE 5.5: Cost function

TABLE	5.3:	Solutions
-------	------	-----------

Design param.	Mean	Std. dev.
WM1,WM2	$81.47 \mu m$	37.99nm
WM3,WM4	$2.97 \ \mu m$	0nm
WM7,WMB	$66.51 \ \mu m$	265.16nm
WM5	$6.03 \mu m$	9 nm
WM6	$28.81 \mu m$	55.34nm
WMR	$7.2 \mu m$	0nm
LM1,LM2	$1.35 \mu m$	0nm
LM3,LM4	180 <i>nm</i>	0nm
LM7,LMB	360nm	0nm
LM5	180 <i>nm</i>	0nm
LM6	180 <i>nm</i>	0nm
LMR	180 <i>nm</i>	0nm
CC	1.3pF	0pF

The results at the end of the execution are summarized in the table 5.3. It can be deduced, for differential evolution solutions are concentrated and those tend to the same value for each optimization parameter.

5.2 Optimal sizing for the OTA-Miller amplifier by NSGA-II Algorithm

This example was developed with the purpose of compare the performance of DE and NSGA-II algorithms in the sizing of analog integrated circuits.

The same circuit of the Section 5.1 was optimized by NSGA-II algorithm. NSGA-II kernel already implemented in C language by Deb, et al. [16], has been used herein, which is sketched in Algorithm 4. The simulation parameters and the specifications are the same of Tables 5.1 and 5.2, respectively. However, for the NSGA-II execution the Gain and

Bandwidth are considered as objectives to maximize, while the other specifications are considered as constraints.



FIGURE 5.6: Comparison between DE and NSGA performance

Figure 5.6 shows the comparison between DE and NSGA-II results. In the DE algorithm the population evolves in order to reach a value for all the specs in table 5.2, specifically for Gain and Bandwidth, the algorithm try to reach 60dB and 100MHz, respectively. Both thresholds are represented in Figure 5.6 as the vertical and horizontal lines. In that way, the solution of the DE algorithm is the intersection point between the lines. In other part, the same figure shows the pareto front (set of solutions) after the NSGA-II execution for 100 and 150 generations. In the case of 100 generations, any of the solutions reach or exceeds both thresholds, it means that for this number of iterations NSGA-II performance is worse than DE. But, because the nature of the NSGA-II, as more iterations better the solutions will be. It is demonstrated when the same algorithm is executed for 150 generations. In this case exist some solutions that exceeds both thresholds; in consequence NSGA-II algorithm can exhibits better performance than DE depending on the number of iterations in the execution.



FIGURE 5.7: Average constraint violation comparison for DE and NSGA-II

Figure 5.7 illustrates the comparison between the average constraint violation for DE and NSGA-II algorithms. As expected, NSGA-II converges faster, it means, this algorithm requires less generations to accomplish all the restrictions. However, it is necessary to

remember that for this example just five of the seven specs are setted as constraints for the NSGA-II based optimization.



FIGURE 5.8: Comparison of the solutions of DE and NSGA-II algorithms

Then, the Figure 5.8 compares the solutions of both algorithms. For DE algorithm the number of solutions is equal to population size, and all of them are almost equal. Nevertheless, NSGA-II number of solutions is less than the population size, each one exhibits different performance (represented in pareto front) and all are different between them.

The execution time for DE algorithm was 699s meanwhile the execution of the same 150 generations takes 728s for the NSGA-II algorithm. The second algorithm is more complex so it is expected this execution time was bigger. However, the most consuming time part of both algorithms is the evaluation step. Due to the evaluation stage employs the same circuit simulations for both algorithms, the two execution times are similar.

5.3 Optimal sizing for the OTA-Miller amplifier by NSGA-II Algorithm comparing real vs integer codification

NSGA-II kernel already implemented in C language by Deb, et al. [16], has been used herein, which is sketched in Algorithm 4.

The Operational Transconductance Amplifier (OTA) shown in Fig. 5.9, consists of 8 MOSFETs. It has two gain stages and a frequency compensation network given by resistor R_z and capacitor C_C . The first stage is composed by a differential input pair (MOSFETs M1 and M2), and a current mirror as load (M3, M4). The second stage (M5, M6) improves the differential gain, reduces the output impedance and also reduces the bandwidth. This is a well-known trade-off called gain-bandwidth (GBW) product. R_z and C_C partially compensate the bandwidth (BW) reduction effect. R_z can be realized with a MOSFET (e.g. M_z) operating in the linear region. I_{bias} and MB generates a voltage to properly bias M6 and M7. For the SPICE simulations, the following conditions were employed: The MOSFETs model is BSIM 3 level 49 for standard IC technology of 180nm, the load capacitance $C_L = 3pF$, $V_{dd} = 3.3V$ and $V_{ss} = 0V$.



FIGURE 5.9: OTA Miller amplifier.

For optimizing this amplifier, two objectives are considered to be maximized, they are: gain and bandwidth. Also four restrictions are considered:

- All the MOSFETs should operate in the saturation region. Except M_z if R_z is realized with a MOSFET,
- Phase margin > 50 degrees,
- Input Offset < 1 mV,
- Common-mode rejection ratio (CMRR) > 50dB.

The chromosome is formed by 14 genes, as listed in Table 5.4. In this manner, the first column lists the gene associated to the design variable given in the second column, and to the respective MOSFET(s) listed in the third column. The fourth and fifth columns list the minimum and maximum values for the integers encoding the associated gene. Those integers will be multiplied by the IC technology declared with the command scale = 90nm, so that they are equivalent to the traditional real encoding values shown at columns sixth and seventh, respectively. The integer encoding values have reduced search spaces, which were established according to the current-branches-bias-assignments approach introduced in [42]. The integer encoding values should change when scaling to a different IC technology.

The NSGA-II algorithm was executed for a population of 100 individuals along 600 generations, with probability of crossover 0.9, probability of mutation 0.5, value of distribution index for crossover 10 and value of distribution index for mutation 20. The two objectives values obtained at the last generation are presented in Fig. 5.10, where one

gene	Design var.	Element	Int. min	Int. max	Real. min	Real. max
x1	W	MB,M6	100	999	9u	89.91u
x2	W	M7	100	999	9u	89.91u
x3	W	M1,M2	100	999	9u	89.91u
x4	W	M3,M4	100	999	9u	89.91u
x5	W	M5	100	999	9u	89.91u
x6	W	Mz	80	200	7.2u	18u
x7	L	MB,M6	2	25	0.18u	2.25u
x8	L	M7	2	79	0.18u	7.11u
x9	L	M1,M2	2	79	0.18u	7.11u
x10	L	M3,M4	2	79	0.18u	7.11u
x11	L	M5	2	79	0.18u	7.11u
x12	L	Mz	2	79	0.18u	7.11u
x13	Capacitance	C_C	1p	20p	1p	20p
x14	Current	Ibias	20u	200u	20u	200u

TABLE 5.4: OTA encoding.

can appreciate the very known trade-off between the gain and bandwidth. The optimization was performed using our proposed integer encoding and also using traditional real encoding. Table 5.5 briefly summarizes the statistical results for the two objective functions (gain and bandwidth (BW)), and three constraints, namely: phase margin (PM), common-mode rejection ratio (CMRR), and voltage offset. Table 5.5 lists the maximum (MAX), minimum (MIN), average (AVG) and standard deviation (STD) values for both integer and real encodings. Additionally, Table 5.6 presents the genes values at the last generation. As one sees, the better performances were computed using the proposed integer encoding. In addition, using integer encoding it is not necessary to perform a post-processing step (wasting computing-time), as when using real encoding values for accommodating or rounding the feasible Ws and Ls to be multiples of the IC technology.



FIGURE 5.10: Pareto Front for sizing the OTA Miller amplifier using integer and real encoding values.

One can execute the sizing optimization process more times changing the number of individuals, the number of generations, as well as the search spaces for the genes encoding

	Integer encoding				Real encoding			
Performances	MAX	MIN	AVG	STD	MAX	MIN	AVG	STD
Gain[dB]	87.19	40.20	60.80	13.59	80.74	40.16	60.42	12.32
BW[KHz]	2191	0.852	609.48	685.58	1967	1.22	529.69	614.23
$PM > 50^{\circ}$	63.21	50.01	53.75	3.38	60.45	50.14	52.93	2.90
CMRR>50dB	96.91	50.18	67.97	13.78	87.43	50.09	65.05	11.43
offset<0.1V	0.0085	-0.003	0.0007	0.0030	0.005	-0.003	0.00030	0.0016

TABLE 5.5: OTA results

gene	Max	Min	Mean	Std. Dev.
x1	986	203	501.333	200.876
x2	976	295	789.125	166.036
x3	694	139	393.333	105.525
x4	998	267	765.083	232.657
x5	302	100	123.812	50.7152
x6	162	95	129.438	15.5466
x7	53	2	10.7917	16.0756
x8	75	2	12.75	18.7906
x9	40	2	7.41667	8.47985
x10	78	48	66.375	8.67284
x11	18	3	6.79167	3.97844
x12	77	14	42.1667	16.4127
x13	6	1	1.60417	1.12495
x14	199	143	186.792	11.2476

TABLE 5.6: Last genes values.

the Ws and Ls, but at the end one can infer that the proposed integer encoding guarantees optimal performance with the advantage of avoiding the post-processing step for rounding-off the feasible sizes solutions, using real encoding, to multiples of the IC technology. It means that the feasible Ws and Ls provided by the proposed integer encoding, can be used directly to generate the layout of the IC. The simulations were executed in an Intel[®] CoreTM i7 @ 2.4GHz processor. During the simulations, it was appreciated that SPICE employed in average 30% less CPU time using integer encoding values than when using real encoding values. Another advantage of using integer encoding is devoted to a reduction in the memory usage, because in a 64-bit processor the integer variables employ 4 bytes, while real (double) variables employ 8.

5.4 Optimal sizing of a CCII by NSGA-II

In this example a second generation current conveyor is optimized and then employed in the implementation of an filter bank and an oscillator. The first generation current conveyor was introduced in 1968 [56], and then it was evolved to the second generation current conveyor (CCII), introduced in 1970 [57]. The generic description of the CCII includes 3-ports named: X, Y and Z, and which electrical characteristics are described by [58],

$$\begin{bmatrix} I_y \\ V_x \\ I_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{bmatrix} \begin{bmatrix} V_y \\ I_x \\ V_z \end{bmatrix} \implies \begin{cases} I_y = 0 \\ V_x = V_y \\ I_z = \pm I_x \end{cases}$$

Figure 5.11 shows an implementation of the CCII using MOSFETs. Its sizing optimization is performed considering two objective functions: The parasitic resistance at port X (R_x) and its associated frequency response. The first objective should be minimized to obtain a low R_x . The second objective associated to the current transfer between ports X and Z should be maximized to obtain a high cut-off frequency f_{ci} . Both objectives were computed for this CCII in [4], but by using traditional real encoding values for the Ws and Ls sizes of the MOSFETs. Besides, for improving the sizing optimization process, we apply our proposed integer encoding procedure. The optimization problem has four restrictions:

- All the MOSFETs should operate in the saturation region,
- Error voltage tracking between ports Y and X (that behave as a voltage follower) < 0.44dB,
- Error current tracking between ports X and Z (that behave as a current follower) $< 0.05 {\rm dB},$
- Offset between ports Y and X < 100mV.

The set-up of the experiment has the same conditions as for the OTA Miller. The population has 100 individuals, and 600 generations. Now the chromosome consists of 9 genes, as shown in Table 5.7, where the first column lists the genes associated to the design variables given in the second column, and to the respective MOSFET(s) listed in the third column. The fourth and fifth columns list the minimum and maximum values for integer encoding, and columns sixth and seventh list the minimum and maximum values of the genes using real encoding values. As it was done for the OTA Miller amplifier, the minimum and maximum values in both integer and real encoding cases have



FIGURE 5.11: Second generation current conveyor (CCII) taken from [4].

gene	Variable	Element	Int. min	Int. max	Real. min	Real. max
x1	W	M5,M6,M11,	100	999	9u	89.91u
		M12,M13				
x2	W	M7,M8,M9,	100	999	9u	89.91u
		M10				
x3	W	M1,M2	100	999	9u	89.91u
x4	W	M3,M4	100	999	9u	89.91u
x5	L	M5,M6,M11	2	80	0.18u	7.2u
		M12,M13				
x6	L	M7,M8,M9,	2	80	0.18u	7.2u
		M10				
x7	L	M1,M2	2	80	0.18u	7.2u
x8	L	M3,M4	2	80	0.18u	7.2u
x9	Current	Ibias	20u	200u	20u	200u

TABLE 5.7: CCII encoding.

been established by considering the current-branches-bias-assignment process described in [42]. Those values can also change when scaling the IC technology.

The feasible values of the two objectives for the last generation are presented in Fig. 5.12. It is worth mentioning that for this case, the parasitic resistance R_x was measured at direct current (DC) conditions. However, in general the impedance value at port X changes according to the frequency variation, for this reason we present the frequency response of three feasible solutions, that are shown in Fig. 5.13, in which one can see the frequency behavior of the X-port parasitic impedance whose value associated to the resistor R_x remains valid from DC (0Hz) near to the computed f_{ci} . In the Pareto front one can note that the feasible solutions using both integer and real encodings have almost the same behavior, but as mentioned above, the integer encoding requires lower SPICE

	Integer encoding				Real encoding			
Performances	MAX	MIN	AVG	STD	MAX	MIN	AVG	STD
Fci[MHz]	530.6	38.12	374.39	143.25	537.90	63.52	390.51	129.43
$\operatorname{Rx}[\Omega]$	270.26	112.24	156.95	45.21	293.19	110.48	156.47	43.60
Voltage[YX] < 0.44dB	-0.11	-0.24	-0.16	0.04	-0.11	-0.24	-0.17	0.03
Current[XZ] < 0.05 dB	0.05	0.048	0.049	0.00033	0.05	0.046	0.049	0.00052
offset[mV]<100mV	1.02	-3.06	-0.52	0.82	0.206	-2.71	-1.23	0.69

TABLE 5.8: CCII results

computing-time because it avoids the post-processing step for rounding the Ws and Ls sizes to multiples of the IC technology when using real encodings.



FIGURE 5.12: Pareto Front for sizing the CCII using integer (black dots) and real (red dots) encoding values.



FIGURE 5.13: Simulated X-port parasitic impedance, whose parasitic resistor R_x remains constant from DC near to f_{ci}

Finally, Table 5.8 lists the statistical values for the feasible solutions. Additionally, Table 5.9 presents the genes values at the last generation, this means Ws and Ls solutions. In this sizing optimization case of study, both integer and real encodings show similar optimized performances, however the proposed integer encoding presents a reduction in CPU-time and memory usage, as discussed in the previous section for optimizing the OTA Miller amplifier.

gene	Max	Min	Mean	Std. Dev.
x1	369	100	128.28	64.3062
x2	659	100	149.447	114.217
x3	999	118	529.86	346.973
x4	999	102	610.333	356.543
x5	52.5	17.9	23.8248	8.76545
x6	48	11	15.3465	7.17358
x7	3.98	2	2.61646	0.701492
x8	3.91	2	2.28905	0.399405
x9	200	198	199.808	0.278477

TABLE 5.9: Last genes values.

In order to test the feasible solutions for the CCII, it is used to implement a universal filter and a single-resistor controlled oscillator (SRCO).

5.4.1 CCII-based filter working in current-mode

According to [5], a current-mode universal filter can be implemented using three dualoutput current conveyors (DOCCIIs), as shown in Fig. 5.14. By applying symbolic analysis as shown by [59], the low-pass, high-pass, band-pass and stop-band responses can be derived, for which the cut-off frequency is given by,

$$w_c = \sqrt{\frac{1}{C_1 C_2 R_1 R_2}} \tag{5.4}$$

To simulate the frequency response of this universal filter, it were selected feasible sizes for the CCII. Afterwards, since the CCIIs should have two outputs, one simply can add more MOSFETs to provide the other required terminal Z(+) or Z(-), i.e. it is performed by extending the current mirrors formed by M5-M6 and M7-M8, as already shown in [60].



FIGURE 5.14: Current-mode universal filter based on DOCCIIs taken from [5].

Element	Value		
R1	$1.71k\Omega$		
R2	$3.17k\Omega$		
R3	$1.71k\Omega$		
$C_1 = C_2$	68pF		
RL	1Ω		

TABLE 5.10: Circuit element values for the universal filter.

The values of the passive elements are presented in Table 5.10. The SPICE simulation results of the four filter responses at 100kHz, are shown in Fig. 5.15. The output currents were measured by adding resistors RL at terminals i_{o1} , i_{o2} and i_{o3} .



FIGURE 5.15: Frequency responses of the universal filter working in current-mode.

5.4.2 CCII-based SRCO

The SRCO can be implemented by modifying the universal filter from Fig. 5.14, it is obtained by eliminating the independent current source i_{in} , and by adding a connection between a new Z(+) terminal of the $CCII_1$ and the X terminal of the $CCII_3$, as already described in [5]. In this case, the oscillation frequency f_o depends on the values of the elements C_1 , C_2 , R_1 and R_2 , which is described by (5.5).

$$f_o = \frac{1}{2\pi} \sqrt{\frac{1}{C_1 C_2 R_1 R_2}} \tag{5.5}$$

For the SPICE simulation, again it was selected one feasible solution for the CCII, and it evolved to get the required DOCCIIs. The values of the passive elements are listed in Table 5.11. The time-response is shown in Fig. 5.16, where one can appreciate the good oscillatory response.

Value		
$1.5k\Omega$		
$3.5k\Omega$		
$3k\Omega$		
24pF		
1Ω		

TABLE 5.11: Circuit element values for simulating the SRCO.



FIGURE 5.16: Time-response of the SRCO computed by SPICE using an optimized CCII.

5.5 Optimal sizing for the Recycling folded cascode (RFC) amplifier

The architecture of Figure 5.17 was reported in [44], as an improvement of the folded cascode transconductance amplifier.



FIGURE 5.17: Recycled OTA.

For this topology all devices are assumed working in saturation region. The NSGA-II algorithm was employed to achieve the specs of 5.12 and to maximize gain and unity-gain frequency.

The algorithm was executed for a population of 100 individuals and for 100 generations. The simulations were executed in a $\text{Intel}^{\mathbb{R}}$ CoreTM i7 @ 2.4GHz processor. The measured specs obtained for the 7 solutions found are shown in Table 5.13.

Element	Value		
Technology	TSMC $0.18 \mu m$		
Input offset[V]	$< 1 \mathrm{mV}$		
Slew Rate[V/ μ s]	> 90		
Open Loop PM	> 70		
Capacitive Load	$5.6 \mathrm{pF}$		
Power consumption	${<}800\mu{ m W}$		

TABLE 5.12: OTA recycling Specs.

TABLE 5.13: OTA recycling Results.

Ind.	Gain[dB]	GBW[MHz]	$\mathbf{PM}[deg]$	$Power[\mu V]$	In. offset[μV]	$\mathbf{SR}+[\mathbf{V}/\mu\mathbf{s}]$	\mathbf{SR} - $[\mathbf{V}/\mu\mathbf{s}]$
1	65.56	86.39	74.99	713	436	262	346.5
2	61.33	108.1	71.59	784	430	199.4	339
3	62.79	90.42	74.01	721	400	275.8	314.6
4	64.2	86.89	75.51	749	431	212.6	342.4
5	66.04	70.47	75.59	745	447	176	304
6	57.16	116.5	72.61	785	386	295.4	319
7	61.99	93.44	71.65	709	400	294.1	299.9



FIGURE 5.18: Recycled OTA solutions.

Figure 5.18 shows the evolution of the average violation constraint, as well as the evolution (increase) of the objectives gain and GBW. From these plots one can conclude on the fast convergence of the algorithm which in 59 generations accomplish all the individuals satisfying all constraints.

5.6 Optimization employing evolutionary algorithms combined with g_m/I_D

5.6.1 Optimal sizing for the OTA Miller amplifier employing DE combined with g_m/I_D

The procedure described on Chapter 3 was applied in the sizing of amplifiers. The conditions for a MOSFET operating in saturation region are:

$$V_{gs} > V_{th} \tag{5.6}$$

$$V_{ds} > V_{gs} - V_{th} \tag{5.7}$$

Furthermore, defining a ΔV voltage due to the overdrive variations tolerance as in [61], the saturation condition can also be expressed as:

$$V_{gs} \ge V_{th} + \Delta V \tag{5.8}$$

$$V_{ds} \ge V_{gs} - V_{th} + \Delta V \tag{5.9}$$

Employing these simple inequations, the biasing voltages V_{DD} and V_{SS} and the desired output voltage swing are finding all the set of conditions that the nodal voltages must satisfy. After that, a sweep for all nodal voltages is made from V_{SS} to V_{DD} in order to find all the nodal voltages combinations that satisfy the saturation biasing conditions.

The collected voltage ranges of nodal voltages are mapped in V_{gs} voltages of all MOS-FETs. Then, the initial extracted MOSFET characterization data is employed to mapping the V_{gs} in the corresponding $\frac{Id}{W}$. Finally, to find the sizes W for each x transistor, it is possible to use a relationship between $\frac{Id}{W}$ and the Ids of the desired transistor, as follows:

After, for sizing a particular analog circuit one should propose an initial guess for the DC operating point. It means biasing voltages and currents with the aim to guarantee that all transistors will be operating in saturation region. In this context, one can suppose reasonable values for overdrive voltages (V_{ov}) and gate to source voltages due V_{DD} and V_{SS} . Additionally, the derived analytical expressions which define the frequency response: gain, and roots (which stablish the Bandwidth and phase margin) for the particular circuit, can be generated. In order to employ the g_m/I_D technique to outline

the parameters range, the following inequalities summarize the bias conditions to keep all transistors are operating in saturation region.

$$\begin{aligned} V_{r1} &\geq V_{ss} + V_{tn} + \Delta V \\ V_{r1} &\leq V_{ocm} + (V_{sw}/2) + V_{tn} + \Delta V \\ V_{r2} &\geq V_{r1} - V_{tn} + \Delta V \\ V_{r2} &\leq V_{icm} - V_{tn} - \Delta V \\ V_{r3} &\geq V_{icm} - V_{tn} + \Delta V \\ V_{r3} &\geq V_{r4} - |V_{tp}| + \Delta V \\ V_{r3} &\leq V_{dd} - |V_{tp}| - \Delta V \\ V_{r4} &\geq V_{icm} - V_{tn} - \Delta V \\ V_{r4} &\geq V_{ocm} + (V_{sw}/2) - |V_{tp}| + \Delta V \\ V_{r4} &\leq V_{dd} - |V_{tp}| + \Delta V \end{aligned}$$

The V_{gs} for each transistor is due to:

$$V_{gs1} = V_{icm} - V_{r2}$$
$$V_{gs3} = V_{dd} - V_{r3}$$
$$V_{gs5} = V_{dd} - V_{r4}$$
$$V_{gs7} = V_{r1} - V_{ss}$$



FIGURE 5.19: Cost evolution.

Figure 5.19, shows the comparison of the cost value evolution during 100 generations. From that Figure, it is clear that to reach the same minimal cost, the DE approach needs 85 generations, meanwhile the $DE + g_m/I_D$ takes only 40 generations. The execution time for the first case reaches the minimum value to be 850.92s, while the execution time for reaching the minimum value with this approach is 474.51s. It means, including the $DE + g_m/I_D$ it is possible to reduce the execution time around 44%.

The final population cost is illustrated in Figure 5.20.



FIGURE 5.20: Final Population Cost.

5.6.2 Optimal sizing for the OTA Miller amplifier employing NSGA-II combined with g_m/I_D

The same procedure previously described was employed to integrate g_m/I_D technique with the optimization algorithm NSGA-II.

Figure 5.21 the average violation constraint for three cases: only NSGA-II and two implementations of NSGA-II $+g_m/I_D$. The implementation (1) is based on the biasing conditions set detailed previously, while in the implementation (2) and specific operational point was proposed, it means a fixed value for each nodal voltage. It was expected for a specific biasing values the search space is even smaller than the based on biasing conditions and it becomes an advantage in terms of convergency speed.



FIGURE 5.21: Violation constraint average comparison.

Figure 5.22 shows the results for the three cases. Based on the picture can be concluded that the pareto front in similar for all the cases in terms of the range achieved for each objective, however as smaller the search space, bigger the solutions set.



FIGURE 5.22: Best population.

5.7 Optimization employing evolutionary algorithms under process variations

The set of solutions shown in Figure 5.22, accomplish all the constraints for nominal conditions of operation. However, the same sizings were simulated using the corner conditions of voltage, temperature and process, and the results are shown in Figures 5.23(a) to 5.24(f).



FIGURE 5.23: Solutions under variations for corners: TT and SS



FIGURE 5.24: Solutions under variations for corners: SF, FS and FF

According with the simulation results, PVT variations make the sizing's constraint violation increase, it means that under variations the sizes can not accomplish the specifications and also the pareto front suffers modifications.

So, by applying the implementation technique formulated on Section 4.2 and based on all corners evaluation, new solutions were obtained. Figure 5.25 shows the violation constraint for the whole execution. Recalling how it works, the first search space is generated employing g_m/I_D design technique, after that the evolutionary algorithm is executed for nominal values of temperature, supply voltage and for the TT process corner, this execution ends when the average violation constraint is equal to zero, it means, when all the individuals of the population reach the constraints. In this case it happens in iteration 28. The plot in red dots represents the violation constraint for the first generations. Then, the search space is again reduced according with the minimum and maximum values for each parameter of the individuals that accomplish the constraints (generation 28). With the new search space the algorithm is executed again but now the violation constraint is the sum of the cost for all 45 corners. This execution is showed in black dotes. As expected the violation constrain reaches bigger values because all corners evaluation, however population evolution leads in less cost values, it means, individuals that reach the constraints despite the PVT variations. The second algorithm execution stops when the maximum number of iterations is reached, in this case 150. This test was made for a population of 150 individuals.



FIGURE 5.25: Violation constraint average comparison.

5.8 Sensitivity Analysis

According with the analysis described on Chapter 4. The sensitivity analysis of the OTA Miller of Figure 5.1 was derived. For that purpose the symbolic expression of the transfer function was derived as a function of all circuit parameters, including g_m , g_{ds} , $c_g s$ and $c_g d$ of each transistor. The differential of transfer function in the frequency domain w.r.t. each parameter is shown in Figure 5.26, for magnitude and 5.27 for phase.

Previously, it is well known that the gain of the amplifier is due to the product of the gain of both stages. The gain of the first one is related with the transconductance of the input pair transistor by the output resistance, which is inversely the output conductance of the input pair and active load transistors. The second stage gain is related directly with the gain transistor and inversely with the output conductance of the load or biasing transistor.

According with magnitude plot is possible to deduce that for low frequencies (near to DC operating) the most sensitive parameters are g_{m5} , g_{ds6} , g_{m2} , g_{ds2} and g_{ds4} .



FIGURE 5.26: Sensitivity of |H(s)| w.r.t each OTA parameter.



FIGURE 5.27: Sensitivity of $\angle H(s)$ w.r.t each OTA parameter

Once the more sensitivity element is identified, a sweep of the sizes associated with the parameter can be done in order to identify the variation percentage that can support while it still accomplish the specifications. As a result, the individuals with the largest variation range can have preference to be selected.

For instance, in this case a sweep of W(M5) was made and the maximum percentage of variation for each individual is ploted in figure 5.28.

In this example, the individuals with larger percentage of variation can ensure more robustness to variations.



FIGURE 5.28: Percentage of variation of W(M5)
Chapter 6

Conclusion

Analog design include a lot of challenges for a designer. All the constraints and the presence of second order effects in a determined fabrication technology make the automatization of the design a difficult task.

Facing this problem, in this work was showed the usefulness of applying optimization techniques for the automatic sizing optimization of analog ICs by applying evolutionary algorithms; specifically the implementation was done for both Differential Evolution and NSGA-II algorithms. As a first improvement to the current approaches, it was stressed a comparison between using traditional real and integer encoding; our proposed integer encoding for ICs, provides feasible Ws and Ls sizes of MOSFETs that can be used directly to generate the layout of the design, because those optimal sizes are multiples of IC technology. In this manner, our proposed integer encoding eliminates the post-processing tasks on rounding-off the feasible solutions using traditional real encoding values. Another important advantage of using integer encoding is the reduction in the memory usage, because in a 64 bit processor a integer variable employs 4 bytes, while a real (double) variable needs 8 bytes.

The proposed integer encoding was used in the optimal sizing of the OTA Miller amplifier and the CCII. The sizing optimization for both ICs using integer encoding showed a reduction of execution time and memory usage vs. using real encoding. The optimized CCII was tested to implement a universal filter and an SRCO, showing good responses, thus justifying the usefulness of the proposed integer encoding.

A new sizing approach that combines the g_m/I_d technique with evolutionary algorithms, has been introduced. The results provided in the Chapter 5 shows that by including the g_m/I_d technique to DE or NSGA-II algorithms, the multi-objective optimization converges in less generations than when using only the optimization algorithm, as traditional cases. The main advantages of this approach are: the searches spaces are feasible values for the given technology, the bias conditions are guaranteed and the computing time required by the algorithm is diminished because the convergence of the algorithm is improved, so the circuit simulator (SPICE, located in the optimization loop) to evaluate the fitness functions, is less times called.

Another apport of this work is the consideration of the variations as a tool to compare the solutions found by the algorithm. Variations in process fabrication, voltage fluctuations and temperature changes modify the response of a circuit. In analog circuits this variations derive in impairing the response features. For that reason, the automatic design tool was modified including the evaluation of 45 PVT corners and minimizing the total cost function. As a result, now the solutions of the algorithm are sizings that accomplish the specifications and also are robust to variations.

Another option to estimate variations was explored. In this work was demonstrated usefulness of performing symbolic sensitivity analysis for compare the impact of each circuit element in the whole behavior of analog circuits. A tool for deriving symbolic expressions for transistor level circuits was developed in this work. After that, an Graph based technique for symbolic differentials was employed to derive the sensitivity with respect of each parameter. The sensitivities were ranked to identify the most sensitive elements, and the procedure was validated by performing HSpice simulations to compare numerical results with the associated symbolic expressions. Finally, from the simulated results, we can conclude that the symbolic sensitivity analysis is an effective method to estimate which components requires special attention in the design process to keep the good performance of an analog IC.

Appendix A

Publications

Adriana Carolina Sanabria-Borbon, Esteban Tlelo-Cuautle, Sizing Analog Integrated Circuits by Combining gm/ID Technique and Evolutionary Algorithms, IEEE 57th Int'l Midwest Symposium on Circuits & Systems (MWSCAS), pp. 234-237, ISBN: 978-1-4799-4132-2, College Station, Texas, August 3-6, 2014.

A.C. Sanabria-Borbon, E. Tlelo-Cuautle, Symbolic sensitivity analysis in the sizing of analog integrated circuits, 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 440-444, ISBN: 978-1-4799-1460-9, México City, Sept 30- Oct 4, 2013.

A.C. Sanabria Borbón, E. Tlelo-Cuautle, Optimización de Filtros Activos y Osciladores
Combinando Verilog-A y SPICE, CIICA 2013-SOMIXXVIII Congreso de Instrumentación,
Sn. Francisco de Campeche, Campeche, México 28-31 de octubre, 2013.

Bibliography

- Matthias Sylvester. Circuit Sizing w / Corner Models Challenges & Applications. MunEDA GmbH, 2013.
- [2] Behzad Razavi. *Design of analog CMOS integrated circuits*. Tata McGraw-Hill Education, 2002.
- [3] Trent McConaghy and Patrick Drennan. Variation-aware custom ic design: Improving pvt and monte carlo analysis for design performance and parametric yield. *Solido White Paper*, 2011.
- [4] Mouna Kotti, Mourad Fakhfakh, and Maria Helena Fino. On the dynamic roundingoff in analogue and RF optimal circuit sizing. *International Journal of Electronics*, pages 1–17, April 2013. ISSN 0020-7217. doi: 10.1080/00207217.2013.784945. URL http://www.tandfonline.com/doi/abs/10.1080/00207217.2013.784945.
- [5] K.K. Abdalla, D.R. Bhaskar, and R. Senani. Configuration for realising a currentmode universal filter and dual-mode quadrature single resistor controlled oscillator. *IET Circuits, Devices & Systems*, 6(3):159, 2012. ISSN 1751858X. doi: 10.1049/iet-cds.2011.0160. URL http://digital-library.theiet.org/content/ journals/10.1049/iet-cds.2011.0160.
- [6] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, David E Goldberg, and John R Koza. Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition Genetic and Evolutionary Computation Series Series Editors Selected titles from this series :. Springer, 2007. ISBN 9780387310299.
- [7] H-O Scheck. Beyond the law. Communications Engineer, 5(2):34–39, 2007.
- [8] Chris Toumazou, George Moschytz, Barrie Gilbert, and Ganesh Kathiresan, editors. *Trade-Offs in Analog Circuit Design*. Kluwer Academic Publishers, Boston, 2002. ISBN 1-4020-7037-3. doi: 10.1007/b117184. URL http://www.springerlink.com/ index/10.1007/b117184.
- [9] Helmut E Graeb. Analog design centering and sizing. Springer, 2007.

- [10] Walter Daems, Georges Gielen, and Willy Sansen. Simulation-based automatic generation of signomial and posynomial performance models for analog integrated circuit sizing. In Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design, pages 70–74. IEEE Press, 2001.
- [11] Frederico Rocha, Ricardo Martins, NunoC Lourenço, and Nuno Horta. State-of-theart on automatic analog ic sizing. In *Electronic Design Automation of Analog ICs combining Gradient Models with Multi-Objective Evolutionary Algorithms*, Springer-Briefs in Applied Sciences and Technology, pages 7–22. Springer International Publishing, 2014. ISBN 978-3-319-02188-1. doi: 10.1007/978-3-319-02189-8_2. URL http://dx.doi.org/10.1007/978-3-319-02189-8_2.
- [12] Sawal Hamid Md Ali. System Level Performance and Yield Optimisation for Analogue Integrated Circuits. PhD thesis, UNIVERSITY OF SOUTHAMPTON, 2009.
- [13] Anyong Qing. Differential Evolution: Fundamentals and Applications in Electrical Engineering. John Wiley & Sons, 2009. ISBN 9780470823927.
- [14] Ivic Guerra Gomez. Optimization of Analog Circuits by Applying Evolutionary Algorithms. PhD thesis, INAOE, 2012.
- [15] Giuseppe Nicosia, Salvatore Rinaudo, and Eva Sciacca. An evolutionary algorithmbased approach to robust analog circuit design using constrained multi-objective optimization. *Knowledge-Based Systems*, 21(3):175–183, April 2008. ISSN 09507051. doi: 10.1016/j.knosys.2007.11.014. URL http://linkinghub.elsevier. com/retrieve/pii/S0950705107001049.
- [16] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation IEEE Transactions on*, 6(2):182–197, 2002.
- [17] Bo Liu, Georges Gielen, and Francisco V. Fernández. Automated Design of Analog and High-frequency Circuits, volume 501 of Studies in Computational Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-39161-3. doi: 10.1007/978-3-642-39162-0. URL http://link.springer.com/10.1007/978-3-642-39162-0.
- [18] Jernej Olenšek, Arpád Bűrmen, Janez Puhan, and Tadej Tuma. DESA: a new hybrid global optimization method and its application to analog integrated circuit sizing. *Journal of Global Optimization*, 44(1):53-77, April 2008. ISSN 0925-5001. doi: 10.1007/s10898-008-9307-9. URL http://link.springer.com/10. 1007/s10898-008-9307-9.

- [19] Nuno Horta. Tutorial on Evolutionary Computation applied to Analog IC Design Tutorial on Evolutionary Computation applied to Analog IC Design.
- [20] I. Guerra-Gomez, E. Tlelo-Cuautle, and Luis G. de la Fraga. Richardson extrapolation-based sensitivity analysis in the multi-objective optimization of analog circuits. *Applied Mathematics and Computation*, 222(0):167 176, 2013. ISSN 0096-3003. doi: http://dx.doi.org/10.1016/j.amc.2013.07.059. URL http://www.sciencedirect.com/science/article/pii/S0096300313008096.
- [21] Xin Li, Le Jiayong, and Lawrence T. Pileggi. Modeling and Optimization, volume 4. now Publishers Inc., Hanover, 2007. ISBN 9781601980564.
- [22] Trent McConaghy, Kristopher Breen, Jeffrey Dyck, and Amit Gupta. Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide. 2013. doi: 10.1007/978-1-4614-2269-3. URL http://www.springerlink.com/index/10. 1007/978-1-4614-2269-3.
- [23] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82-117, July 2013. ISSN 00200255. URL http://linkinghub.elsevier.com/retrieve/pii/S0020025513001588.
- [24] BV Babu and M Mathew Leenus Jehan. Differential evolution for multi-objective optimization. In *Evolutionary Computation*, 2003. CEC'03. The 2003 Congress on, volume 4, pages 2696–2703. IEEE, 2003.
- [25] Rainer Storn. On the usage of differential evolution for function optimization. In Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, pages 519–523. IEEE, 1996.
- [26] Daniela Zaharie. A Comparative Analysis of Crossover Variants in Differential Evolution. In International Multiconference on Computer Science and Information Technology, pages 171–181, 2007.
- [27] Youyun Ao and Hongqin Chi. Experimental Study on Differential Evolution Strategies. 2009 WRI Global Congress on Intelligent Systems, pages 19-24, 2009. doi: 10.1109/GCIS.2009.31. URL http://ieeexplore.ieee.org/lpdocs/epic03/ wrapper.htm?arnumber=5209142.
- [28] Godfrey Onwubolu and Donald Davendra. Scheduling flow shops using differential evolution algorithm. *European Journal of Operational Research*, 171(2): 674-692, June 2006. ISSN 03772217. doi: 10.1016/j.ejor.2004.08.043. URL http://linkinghub.elsevier.com/retrieve/pii/S0377221704006599.

- [29] Johanna Aalto and Jouni Lampinen. A mutation adaptation mechanism for Differential Evolution algorithm. 2013 IEEE Congress on Evolutionary Computation, pages 55-62, June 2013. doi: 10.1109/CEC.2013.6557553. URL http: //ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6557553.
- [30] Janez Brest, Saso Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *Evolutionary Computation, IEEE Transactions on*, 10(6):646–657, 2006.
- [31] Jason Teo. Exploring dynamic self-adaptive populations in differential evolution. Soft Computing, 10(8):673-686, August 2005. ISSN 1432-7643. doi: 10.1007/s00500-005-0537-1. URL http://link.springer.com/10.1007/s00500-005-0537-1.
- [32] Li Gao-yang and Liu Ming-guang. The Summary of Differential Evolution Algorithm and its Improvements. In 3rd international Conference on Advanced Computer Theory and Engineering (iCACTE) The, number i, pages 153–156, 2010. ISBN 9781424465422.
- [33] Tea Robič and Bogdan Filipič. Demo: Differential evolution for multiobjective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 520–533. Springer, 2005.
- [34] Miloslav KUBAŘ and Jiří JAKOVENKO. A Powerful Optimization Tool for Analog Integrated Circuits Design. RADIOENGINEERING, 22(3):921–931, 2013.
- [35] Biao Luo, Jinhua Zheng, Jiongliang Xie, and Jun Wu. Dynamic Crowding Distance-A New Diversity Maintenance Strategy for MOEAs. 2008 Fourth International Conference on Natural Computation, pages 580–585, 2008.
- [36] Aravind Seshadri. A fast elitist multiobjective genetic algorithm: nsga-ii. Evolutionary Computation, IEEE Transactions on, 6(2):1–4, 2002.
- [37] Manuel Barros, Jorge Guilherme, and Nuno Horta. Analog circuits optimization based on evolutionary computation techniques. *Integration, the VLSI Journal*, 43 (1):136-155, January 2010. ISSN 01679260. doi: 10.1016/j.vlsi.2009.09.001. URL http://linkinghub.elsevier.com/retrieve/pii/S0167926009000406.
- [38] Ivic Guerra-Gomez and E. Tlelo-Cuautle. Sizing Analog Integrated Circuits by Current- Branches-Bias Assignments with Heuristics. *ELEKTRONIKA IR ELEK-TROTECHNIKA*, 19(10):81–86, 2013.

- [39] David M. Binkley. Tradeoffs and Optimization in Analog CMOS Design. 2008. ISBN 9780470031360. URL http://onlinelibrary.wiley.com/book/10.1002/ 9780470033715;jsessionid=898B7D41CF13E149034C45900F8DAC5B.f04t04.
- [40] F Silveira, D Flandre, and P G A Jespers. A gm/ID Based Methodology for the Design of CMOS Analog Circuits and Its Application to the Synthesis of a Siliconon-Insulator. SOLID-STATE CIRCUITS, IEEE, 31(9):1314–1319, 1996.
- [41] Paul Jespers. The gm/ID Methodology, A Sizing Tool for Low-voltage Analog CMOS Circuits. Springer US, Boston, MA, 2010. ISBN 978-0-387-471006. doi: 10.1007/978-0-387-47101-3. URL http://link.springer.com/10.1007/
 978-0-387-47101-3.
- [42] I. Guerra-Gomez and E. Tlelo-Cuautle. Sizing analog integrated circuits by currentbranches-bias assignments with heuristics. *Electronics and Electrical Engineering*, 2014. ISSN 2029-5731.
- [43] Ivick Guerra-Gomez, Trent McConaghy, and Esteban Tlelo-Cuautle. Operating-point driven formulation for analog computer-aided design. Analog Integrated Circuits and Signal Processing, 74(2):345-353, December 2012. ISSN 0925-1030. doi: 10.1007/s10470-012-9998-1. URL http://link.springer.com/10.1007/s10470-012-9998-1.
- [44] Rida S Assaad and Jose Silva-Martinez. The recycling folded cascode: a general enhancement of the folded cascode amplifier. *Solid-State Circuits, IEEE Journal of*, 44(9):2535–2542, 2009.
- [45] Colin C Mcandrew, Ik-sung Lim, Brandt Braswell, and Doug Garrity. Corner Models: Inaccurate at Best, and it Only Gets Worst ... In *Custom Integrated Circuits Conference (CICC)*, pages 13–16, 2013. ISBN 9781467361460.
- [46] Mourad Fakhfakh, Esteban Tlelo-Cuautle, and Francisco V Fernández. Design of analog circuits through symbolic analysis. Bentham Science Publishers, 2012.
- [47] Dimitri De Jonghe and Georges Gielen. Characterization of analog circuits using transfer function trajectories. *Circuits and Systems I: Regular Papers, IEEE Trans*actions on, 59(8):1796–1804, 2012.
- [48] Cristian Ferent and Alex Doboli. Symbolic matching and constraint generation for systematic comparison of analog circuits. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 32(4):616–629, 2013.
- [49] Guoyong Shi. Graph-pair decision diagram construction for topological symbolic circuit analysis. IEEE Trans. on CAD of Integrated Circuits and Systems, 32(2): 275–288, 2013.

- [50] Marian PierzchaLa and Mourad Fakhfakh. Transformation of lc-filters to active rccircuits via the two-graph method. *Microelectronics journal*, 42(8):999–1005, 2011.
- [51] Santiago Rodriguez-chavez and Esteban Tlelo-cuautle. Graph-Based Symbolic Technique for Improving Sensitivity Analysis in Analog Integrated Circuits.
- [52] Carlos Sanchez-Lopez. Pathological equivalents of fully-differential active devices for symbolic nodal analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions* on, 60(3):603–615, 2013.
- [53] Franco Maloberti. Analog design for CMOS VLSI systems. Springer, 2001.
- [54] Guoyong Shi and Xiaoxuan Meng. Variational analog integrated circuit design via symbolic sensitivity analysis. 2009 IEEE International Symposium on Circuits and Systems, (4):3002-3005, May 2009. doi: 10.1109/ISCAS.2009.5118434. URL http: //ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5118434.
- [55] Nuno Lourenço and Nuno Horta. Genom-pof: Multi-objective evolutionary synthesis of analog ics with corners validation. In Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, GECCO '12, pages 1119–1126, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1177-9. doi: 10.1145/2330163.2330318. URL http://doi.acm.org/10.1145/2330163.2330318.
- [56] K. C. SMITH and A. SEDRA. The current conveyor—A new circuit building block. Proceedings of the IEEE, pages 3–4, 1968.
- [57] A. Sedra and K. Smith. A Second-Generation Current Conveyor and Its Applications. *IEEE TRANSACTIONS ON CIRCUIT THEORY*, (February):132–134, 1970.
- [58] Giuseppe Ferri and Nicola C. Guerrini. Low-Voltage Low-Power CMOS Current Conveyors. Kluwer Academic Publishers, Boston, 2004. ISBN 1-4020-7486-7. doi: 10.1007/b105853. URL http://www.springerlink.com/index/10.1007/b105853.
- [59] Carlos Sánchez-lópez. Pathological Equivalents of Fully-Differential Active Devices for Symbolic Nodal Analysis. 60(3):603–615, 2013.
- [60] Miguel Aurelio Duarte-Villaseñor, Esteban Tlelo-Cuautle, and Luis Gerardo Fraga. Binary Genetic Encoding for the Synthesis of Mixed-Mode Circuit Topologies. *Circuits, Systems, and Signal Processing*, 31(3):849–863, September 2011. ISSN 0278-081X. doi: 10.1007/s00034-011-9353-2. URL http://link.springer.com/10.1007/s00034-011-9353-2.

[61] Cheng-Wu Lin, Pin-Dai Sue, Ya-Ting Shyu, and Soon-Jyh Chang. A bias-driven approach for automated design of operational amplifiers. In VLSI Design, Automation and Test, 2009. VLSI-DAT'09. International Symposium on, pages 118–121. IEEE, 2009.