Optimization of CMOS OTAs applying metaheuristics

by

Martín Alejandro Valencia Ponce

Thesis submitted as a partial requirement to obtain the degree of **Master in science in Electronics Area** at the National Institute of Astrophysics, Optics and Electronics

Supervised by:

Dr. Esteban Tlelo Cuautle, INAOE

Optimization of CMOS OTAs applying metaheuristics

Master's Thesis

BY:

Martín Alejandro Valencia Ponce

ADVISORS:
DR. ESTEBAN TLELO CUAUTLE

National Institute of Astrophysics, Optics and Electronics
Electronics Coordination

TONANTZINTLA, PUEBLA. AUGUST 2019

Acknowledgements

First of all, I thank God and the Virgen María for giving me the strength to get ahead in any situation.

I would like to thank my advisor, Dr. Esteban Tlelo Cuautle for his support and help throughout the project.

I am deeply grateful to my mother Ortencia Ponce Llamas for her unconditional support throughout my life and her effort to give me the opportunity to do what I like. For being a wonderful mother and the person responsible for most of my achievements.

To my brother Carlos Fernando Gonzalez Ponce for his patience and advice.To my girlfriend Perla Rubi Castañeda Aviña for her support and patience at all times and José Luis Palacios Mejía for his wise advice.

I would like to express my sincere gratitude to the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) and the Consejo Nacional de Ciencia y Tecnología (CONACyT) for its support.

Resumen

El diseño de circuitos integrados (CIs) analógicos se ha convertido en un proceso complejo, debido al gran numero de parámetros, variables de diseño y compromisos entre las características eléctricas de cada circuito. Por lo tanto, se han implementado diferentes técnicas de optimización para el diseño de CIs, una de las más utilizadas actualmente es la aplicación de metaheuristicas para el dimensionamiento de CIs, ya que proporcionan un conjunto de soluciones factibles de acuerdo al parámetro a optimizar. En esta Tesis se aplican metaheuristicas mono y multi-objetivo para el dimensionamiento de Amplificadores Operacionales de Transconductacia (OTAs) con el objetivo de minimizar el área de silicio y garantizar que los transistores MOS operen dentro de un punto de operación de corriente directa (DCOP) adecuado.

En esta Tesis se describen los algoritmos de optimización basados en poblaciones y en algoritmos evolutivos como Optimización de Enjambre de Partículas (PSO), Muchos Enlaces de Optimización (MOL) y Algoritmo Genético de Clasificación No dominado-II (NSGA-II) aplicados al dimensionamiento de los OTAs Miller y Recycled Folded Cascode, utilizando las tecnologías UMC de 180nm y onsemiconductor de 500nm, respectivamente. Además, se realiza el diseño y la caracterización de un OTA Miller utilizando las ecuaciones del modelo cuadrático. Posteriormente se describe el procedimiento que se llevo a cabo para estimar el área de silicio de cada OTA y las condiciones para garantizar que los transistores MOS operen dentro de la región de inversión fuerte.

Se muestra el comportamiento y los resultados de las partículas de los algoritmos PSO y MOL a traves de las generaciones, y los frentes de Pareto correspondientes al algoritmo NSGA-II. Finalmente, de acuerdo con los resultados obtenidos de los algoritmos para el dimensionamiento de los OTAs, el layout y las simulaciones post-layout se llevan a cabo para verificar que los circuitos optimizados sean robustos a variaciones de proceso, voltaje y temperatura (PVT).

Abstract

Analog integrated circuits design has been a challenge, due to the large number of parameters, design variables and trade-offs among the electrical characteristics of each circuit. Therefore, different optimization techniques have been implemented for the design of ICs. One of the most used techniques is the application of metaheuristics for the sizing of ICs, since they provide a set of feasible solutions according to the parameter to be optimized. In this Thesis mono and multi-objective metaheuristics are applied for the sizing of Transconductance Operational Amplifiers (OTAs) with the aim of minimizing the silicon area and guaranteeing that the MOS transistors operate within a suitable direct current operating point (DCOP).

In this Thesis are described population-based optimization algorithms and evolutionary algorithms, such as Particle Swarm Optimization (PSO), Many Optimization Liaisons (MOL) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) applied to the sizing of both the Miller and Recycled Folded Cascode OTAs, using UMC 180nm and onsemiconductor 500nm technologies, respectively. Moreover, the design and characterization of a Miller OTA is performed using the quadratic model. Subsequently, the procedure that was carried out to estimate the silicon area of each OTA and the conditions to ensure that MOS transistors operate within the strong inversion region is described.

The behaviors and results of the PSO and MOL particles through the generations and the Pareto fronts corresponding to the NSGA-II algorithm are shown. Finally, according to the results obtained with the algorithms for the sizing of the OTAs, the layout and the post-layout simulations are carried out to verify that the optimized circuits are robust to process, voltage and temperature variations (PVT).

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Integrated circuit (IC) design using complementary metal-oxide semiconductor (CMOS) technology is a complex process since the performance of the circuit depends on the size of the fabrication process. There are mathematical models that describe the circuit's behavior at different technologies, but usually they do not take into account secondary effects like channel length modulation, body effect and velocity saturation that can affect the circuit's performance. For example, as mentioned in [6], to carry out a good design is necessary to start from a set of previously established specifications. However, there is no general method for IC design, but as mentioned in [1], top-down design methodologies are proposed with the intention of separating a complex problem into several ones with a minor difficulty. In a hierarchical design approach, generally the first stage is the system level, which is based on developing the architecture to satisfy the established specifications, programs such as Matlab and Verilog AMS allow to verify the system's performance. Subsequently, the system is partitioned into blocks, which describe the system's operation through a behavioral programming language such as Simulink, Verilog and VHDL. The next stage corresponds to the circuit level, at this stage the designer requires experience to choose the appropriate topology and his ability to carry out the devices sizing corresponding to the circuit's parameters. There are different methods to implement the transistor sizing, and they require different kind of models such as the quadratic one that is based on independent equations that describe the transistor's behavior, gm/id technique and other automation tools.

1.1. Analog Integrated Circuit Design

The integration of mixed digital-analog circuits in VLSI (Very-Large-Scale Integration) technology is a challenge for analog IC design. For example, as already mentioned in [7], automated synthesis or sizing methodologies are needed to avoid long design times, high design complexity, high cost and highly skilled designers. In practice, and as shown in Fig. 1.1, parameters such as noise, power dissipation, input and output

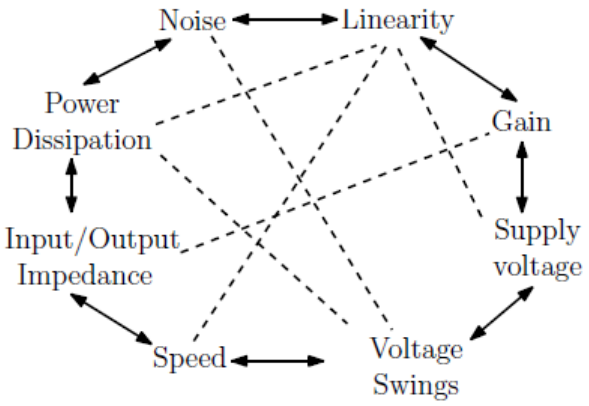


Figure 1.1: Analog design octagon [2]

impedance, linearity, gain, voltage range, speed and voltage sources present a trade-off among them. One example on understanding trade-offs is the one related to have a better gain in an amplifier design, which main condition requires a high output impedance. Other examples of trade-offs can be inferred looking at Fig. 1.1, from which one can conclude that according to [2], analog integrated circuit design is a kind of a multi-dimensional optimization problem. This allows the integrated circuit design process to be automated and optimized, with the aim of minimizing design errors according to target specifications.

The problem of designing an electronic system from its most abstract descriptions can be solved by adopting hierarchical design approximations. For example: in Fig. 1.2 one can see the steps to achieve a CMOS design at different levels of abstraction. The most abstract step is selecting a topology that fits the specifications, this topology can be selected from designer's experience, database or with heuristics rules. Once it was chosen the best topology is important to propose current branches and biasing voltages for each circuit element to start with the next step, transistor sizing. In this step, the width (W) and the length (L) of each transistor are determined in order to find the best sizes that accomplish target specifications. Afterwards, the verification of the circuit performance, can be performed by using circuit simulators like SPICE (Simulation Program with Integrated Circuit Emphasis). If the circuit doesn't complies with all previously established requirements the previous steps are reviewed, otherwise the next step is the layout generation, verification and extraction.

The sizing of analog integrated circuits is not a trivial task, so that one can find different works proposing different sizing techniques because in the analog domain, every topology may require different approaches. According to [1], one can identify two main strategies for performing the automatic sizing of analog ICs: knowledge based and optimization based. The first one is based on calculating the design parameters according to the circuit's characteristics by means of design equations, combining mathematical techniques with the intuition and experience of the designer. The latter consists of

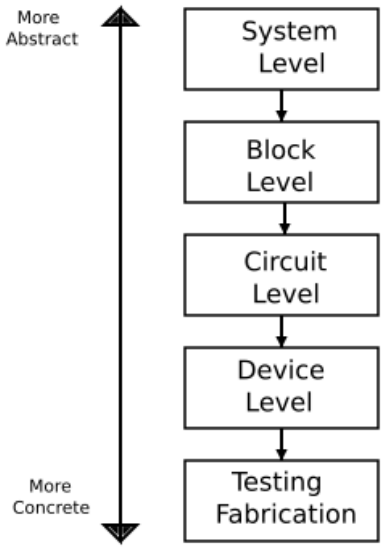


Figure 1.2: Hierarchical approach of ICs design [1]

applying conventional optimization approaches or metaheuristics, and in both cases the optimization loop evaluates circuit performances using a circuit analysis tool like SPICE.

1.2. Metaheuristics

The first appearance of the heuristics was during the second world war in 1940 with Alan Turing who called his method as heuristics search, as mentioned in [11]. After this great contribution the next big step was the development of evolutionary algorithms by John Holland and his collaborators among 1960s and 1970s. In 1975 Kenneth De Jong showed the potential of genetic algorithms for objective functions, whether noisy, multimodal or even discontinuous. The most important time in the development of metaheuristics was in the decade of the 90s with Tabu Search [12] by Fred Glover's in 1997. Then, in 1992, search techniques based on swarm intelligence ants using pheromones to communicate between them, was developed in [13]. Another great advance in the development of metaheuristics was particle swarm optimization (PSO) that is inspired by the social behavior of fish and birds PSO was developed by James Kennedy, and engineer Russell C. Eberhart [14].

The word metaheuristics comes from $meta$ which means the highest level and $heuristic$ which mean the art of discovering new strategies [3]. According to [15], metaheuristics can be classified into two groups: single-solution-based and population-based. The first one is based on a single solution at any time such as search-based (SA). The population-

based metaheuristics are updated iteratively until the termination condition is satisfied and are exploration-oriented, moreover are categorized into evolutionary algorithms (EAs) and swarm-based algorithms.

Another type of classification is presented in [16], where metaheuristics are classified into four categories: relaxation, constructive, search and evolutionary metaheuristics.

- Relaxation metaheuristics propose models that can be simplified to eliminate or weaken a complex problem.
- Constructive metaheuristics are based on procedures that obtain a solution from the analysis and gradual selection of the components that form it.
- Search metaheuristics guide the procedures that use movements to go through the space of alternative solutions and exploit the associated structures.
- Evolutionary metaheuristics are focused on procedures based on sets of solutions that evolve over the solution space.

Optimization has become an important subject of interest for many researchers in order to maximize or minimize the characteristics of some process. According to [11] the vast majority of optimization techniques are metaheuristics, due to its ability to produce feasible solutions to complex problems in a short time. In general, metaheuristics algorithms are based on some abstractions of nature, arguing that nature has found good solutions to all kinds of problems over the years. The most important characteristics of the metaheuristics, is the selection process which ensures that the solutions converge at one point, on the other hand is the randomness that ensures that the algorithm is not stuck in a limited search space.

Metaheuristics allow to solve a problem by delivering satisfactory solutions in a short time. There is no guarantee to find global optimal solutions. Fig. 1.3 shows the genealogy of metaheuristics. Is also observed that since the 80s the development of metaheuristics has exponentially increased due to the demand to improve the optimization process. According to [3] a metaheuristic must take into account two contradictory criteria: exploration of the search space (diversification) and exploitation of the best solutions found (intensification). In one hand the diversification visits unexplored regions to be sure that all the search regions are evenly explored. On the other hand intensification is based on exploring the most promising regions with the aim of finding better solutions.

Due to different optimization methods, metaheuristics can be classified according to certain criteria [3] such as Nature and Nonnature inspired: a large number of metaheuristics are based on a natural process such as evolutionary algorithms and particle swarm optimization, moreover another type of metaheuristics employ physics principles like the simulated annealing. There are also methods of memory usage and memoryless. In the latter, no information extracted is used during the search, an example of this type of local search is greedy adaptive search procedure (GRASP) [17], on the contrary, metaheuristics like tabu search occupies short and long term memory space.

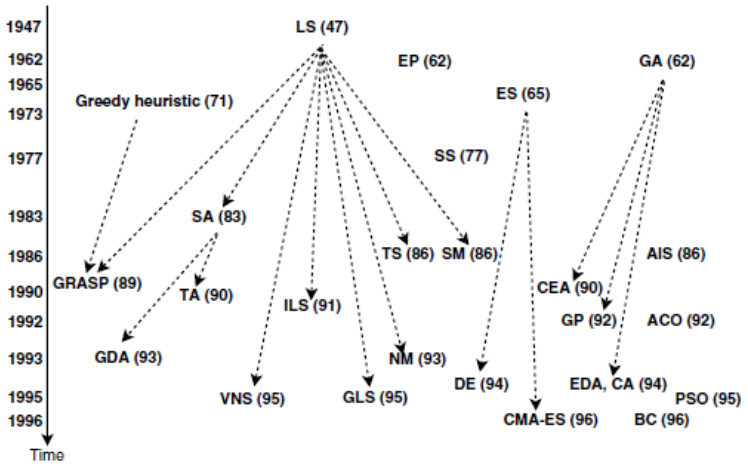


Figure 1.3: Genealogy of the metaheuristics [3]

Also the metaheuristics can be classified in deterministic and stochastic: The former metaheuristics solves an optimization problem by making deterministic decisions while a stochastic metaheuristics applies random rules during the search. Finally another type of classification can be iterative and greedy: Iterative algorithms are initialized with a number of solutions which will be transformed in each iteration using some search operators. Greedy algorithms start from an empty solution, and at each step a decision variable of the problem is assigned.

In recent years, metaheuristics have been applied to the design of integrated circuits, in [18] an evolutionary approach based on a variation of the PSO is implemented with the purpose of sizing a two-stage comparator (TSC) and a CMOS Folded Cascode Operational Transconductance Amplifier (FCOTA) for low power applications. In [19] a highly effective hybrid sizing methodology for the detection of parasitics is proposed, using mono-objective evolutionary algorithms. Another metaheuristic strongly used in transistor sizing is the genetic algorithm as shown in [20], with the purpose of sizing an operational amplifier since it can be translated into a multi-objective optimization task. In [10] is shown the usefulness of assigning current-branches-bias levels, in order to improve the sizing optimization of analog integrated circuits. In addition metaheuristics have been used for multi-objective optimization problems as shown in [21] where the authors use NSGA-II for the sizing of a Recycled Folded Cascode Operational Transconductance Amplifier (RFCOTA). An optimization system which uses the multi-objective evolutionary algorithm based on decomposition (MOEA/D) is presented in [22] for sizing second generation current conveyors (CCIIs). PSO and NSGA-II, have been applied in optimizing CMOS amplifiers, but they require circuit evaluators that embed mathematical models of the MOS transistors.

1.3. MOS Transistor Models

The design of analog and digital ICs can be done using CMOS technologies, but it is important to have a reliable MOS transistor model. These models allow the designer to predict the behavior of MOS transistors under a particular operation condition, since it describes the physical phenomena present in MOS transistors with a small number of physical variables. The basic idea of modeling a device is to have a list of the electrical variables, whether geometrical or physical, in order to identify which ones can function as design parameters. There is a trade-off between the quality and the complexity of the model. Therefore a simple model is generally necessary to facilitate the design and the symbolic hand manipulation like quadratic model. There are many models, like EKV model, with greater complexity that are generally used for computer simulation of circuits employing CMOS devices. Unlike analog design, digital circuits can be modeled as switches, which is a relatively simple model, while analog circuits require a more detailed model. According to [23] a MOS transistor model must take into account the voltage-current relationships, noise and temperature variations and the dynamic behavior of the CMOS devices.

1.3.1. BSIM3v3

BSIM3v3 is the latest industry-standard MOSFET model for deep-submicron digital and analog circuit designs from the BSIM Group at the University of California at Berkeley. BSIM3v3.2.2 is based on its predecessor, BSIM3v3.2. This model is used in the circuit simulator SPICE, its main advantages are:

- Elimination of small negative capacitance values (Cgs, Cgd) in the accumulation-depletion regions.
- Improved modeling of C-V characteristics at the weak-to-strong inversion transition.
- Option of using C-V inversion charge equations of CAPMOD=0,1,2,3 to calculate the thermal noise when NOIMOD=2 or 4.
- A separate set of length/width dependence parameters for the CV model.
- Additional parameter checking.
- Improvements in numerical stability.

According to [24], the implementation of the BSIM3v3 model is performed with 21 files that are divided into 5 parts. 1) Data Structures: BSIM3v3 model is described with pointers to function, which provide specific operations and tables that describe the BSIM3v3 model parameters. BSIM3v3 model needs two specific internal data structures, one for the global device model and the other for defining the individual devices in a circuit. Data placed in the model data structure is static, this data structure contains only the data that is universal to all the devices. 2) Input Routines: These routines are used by pass the input parameters to the device. This routine requires two files,

the first one b3par.c is used to take the parameter values from the input parser and to set the appropriate field in the data structure of the device. Other file is b3mpar.c and provides values for model parameters. 3) Output Routines: This routine include two files b3ask.c and b3mask.c and are used to obtain data from the BSIM3v3 model by the simulator. 4) Structure decomposition routines: These routines are used to separate the data structures. It consists of 3 files, the first file b3dest.c releases all the used memory. b3mdel.c file is designed to delete BSIM3v3 from the circuit. Finally b3del.c is used to delete specified instance from the circuit. 5) Processing Routines: This routine include ten files, the first one is b3set.c file and is used to prepare BSIM3v3 for simulation; b3temp.c completes the parameter preprocessing; b3check.c examines if the values of the parameters are invalid or unreasonable before they are loaded by the simulator for any calculation; b3getic.c is used to convert node initial conditions to device initial conditions; b3ld.c is responsible for evaluating all instances at each iteration in the DC and transient analyses and for loading the Jacobian matrix and right hand side vector with the appropriate values; b3trunc.c is used to compute the truncation error for each device in the circuit; b3cvtest.c performs the necessary convergence testing; b3acld.c is used when ac analysis is performed; b3pzld.c evaluates the conductance at the complex frequency; b3noi.c evaluates all of the noise sources.

This model like others is developed for expressions to different device operation regimes, such as subthreshold and strong inversion. These expressions describe in a very precise way the behavior of the device in its respective region of operation. But in regions of transition these equations are no longer accurate and are practically not a good approximation of the device's behavior. Therefore to correct this error, an unified model should be synthesized to ensure the continuities of current and conductance and their derivatives in all transition regions.

The expressions (1.1) and (1.2) are the charge density in subthreshold and strong inversion, respectively. Both expressions are valid for small V_{DS}

$$Q_{inv} = Q_O exp(\frac{V_{GS} - V_T}{nU_T}) \tag{1.1}$$

$$Q_{inv} = C_{ox}(V_{GS} - V_T) \tag{1.2}$$

Where n is the subthreshold swing parameter and the thermodynamic voltage U_T which is equivalent to $(k * T)/q$ where k is the Boltzman constant, q the elementary charge of the electron and the temperature $T = 300K$. U_T is approximately $25.8mV$. To obtain a unified expression a V_{gsteff} function is introduced to describe the channel charge characteristics from subthreshold to strong inversion.

$$V_{gsteff} = \frac{2nU_T ln[1 + exp(\frac{V_{GS} - V_T}{2nU_T})]}{1 + 2nC_{ox}\sqrt{\frac{2\Phi_s}{q\epsilon_{si}N_{ch}}}exp(-\frac{V_{GS} - V_{TH} - 2V_{off}}{2nU_T})} \tag{1.3}$$

V_{off} is an important parameter which determines the drain current at $V_{GS} = 0$, N_{ch} is the doping concentration in the channel and ϵ_{si} is the silicon's permittivity. Then the unified expression for the channel charge density for subthreshold and inversion region is given by (1.4).

$$Q_{inv} = C_{ox} V_{gsteff} \tag{1.4}$$

The V_{gsteff} function gives a unified expression for the linear drain current. In order to obtain a generalized expression a smooth function of V_{DS} is introduced. In (1.5) is shown the final current equation for both linear and saturation regions.

$$I_D = \frac{I_{dso(vdsat)}}{1 + \frac{R_{ds} I_{dso(vdsat)}}{V_{dsat}}} (1 + \frac{V_{ds} - V_{dsat}}{V_A})(1 + \frac{V_{ds} - V_{dsat}}{V_{ASCBE}}) \tag{1.5}$$

where V_A is the early voltage and R_{ds} is the resistance that is formed between the drain terminals and the source.

1.3.2. Quadratic Model

Nowadays the quadratic model is the most used model due to the few physical parameters that describe the transistor's behavior in each of its operating regions, such as cutoff, linear or triode and saturation. Is important to mention that each of these regions are governed by independent equations.

The derivation of the quadratic model like the EKV model requires an expression for the charge density. From $Q = CV$ is noted that C is the gate capacitance per unit length and V is the voltage difference between gate and channel and Q is the charge density. The gate capacitance is denoted by a capacitance per unit area C_{ox}, taking into account the transistor's width W and also considering that $V = V_{GS} - V_T$, because no mobile charge exists for $V_{GS} < V_T$. Therefore, the charge density is given by (1.6).

$$Q = W C_{ox} (V_{GS} - V_T) \tag{1.6}$$

Then as the channel voltage varies along the length of the transistor is necessary to denote the channel potential at x by $V(x)$, which goes from zero to V_D

$$Q(x) = W C_{ox} (V_{GS} - V(x) - V_T) \tag{1.7}$$

Once an expression for the charge density is obtained, is important to derive an expression for the drain current where basically two important points are considered according to [25]. The first one mentions that I is given by the total charge that passes through the cross section of the bar in one second. The second consideration says that if the carriers move with a velocity vm/s , then the charge enclosed in v meters along the bar and passes through the cross section in one second is equal to Qv, this is expressed in (1.8).

$$I = Qv \tag{1.8}$$

where $v = \mu\frac{dV}{dx}$. Combining (1.7), (1.8) and the definition of v the drain current is (1.9).

$$I_D = WC_{ox}(V_{GS} - V(x) - V_T)\mu\frac{dV}{dx} \tag{1.9}$$

To find an expression of the current in terms of the voltage terminals both sides of (1.9) are integrated

$$I_D = \frac{1}{2}\mu C_{ox}\frac{W}{L}[2(V_{GS} - V_{TH})V_{DS} - V_{DS}^2] \tag{1.10}$$

A MOS transistor is composed of a plate called gate and two junctions called source and drain. The operating principle of a MOS transistor is to apply a voltage on the gate so that current flows from source to drain. The physical effects and derivations of the equations are detailedly explained in [6] and [25]. The operating regions can be derived from the voltages applied to the terminals of the MOS transistor and the drain current (1.10). So if $V_{GS} < V_T$ is said that the MOS transistor is in cutoff region. When $V_{GS} > V_T$ and $V_{DS} < V_{GS} - V_T$ the MOS transistor operates in triode region. In this region the drain current I_D is given by 1.11.

$$I_D = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{GS} - V_T - \frac{V_{DS}}{2})V_{DS} \tag{1.11}$$

In triode region the drain current (1.11) is proportional to V_{GS} and V_{DS}. The MOS transistor operates in saturation region if $V_{GS} > V_T$ and $V_{DS} > V_{GS} - V_T$. In this region the channel is pinched-off and I_D is proportional to $(V_{GS} - V_T)^2$ as shown in (1.12).

$$I_D \approx \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{DSAT})^2(1 + \lambda V_{DS}) \tag{1.12}$$

Figure 1.4 shows the $I_D - V_{DS}$ characteristics. According to [25] the formation of the channel is actually a gradual effect and the device presents a small current when $V_{GS} < V_T$, this effect is called subthreshold conduction and occurs when the MOS transistor changes from cutoff to triode region. Another important consideration is that the gate is isolated from the substrate by means of an insulator layer and the current at the gate I_G is nearly 0.

1.4. Problem Formulation

The design of integrated circuits (ICs) using complementary metal-oxide-semiconductor (CMOS) is a complex process due to the many target specifications, the many design

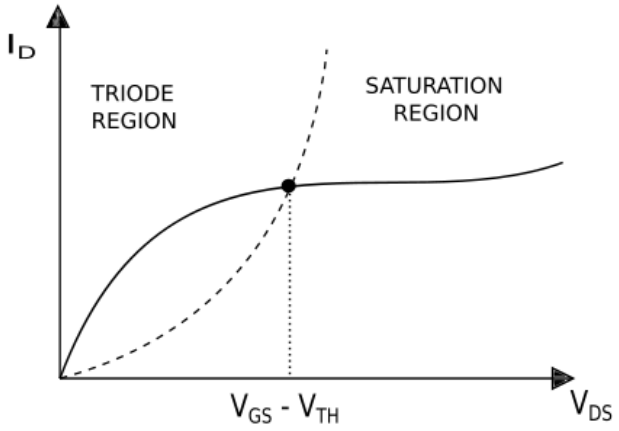


Figure 1.4: $I_D - V_{DS}$ Characteristics

variables, constraints and the sparse ranges of the search spaces of every performance. Therefore analog design automation tools are required to improve the IC design.

Metaheuristics have shown their usefulness in optimizing analog ICs because they link circuit simulators like SPICE to evaluate electrical characteristics that are associated to a CMOS technology. As demonstrated in [26] and [21] metaheuristics like PSO, MOL and NSGA-II present a satisfactory behavior in the transistor sizing. According to the problem, metaheuristics can optimize mono-objective, multi-objective and many-objective.

Although metaheuristics provide feasible solutions, they do not guarantee that the MOS transistors work in the desired direct current operating point (DCOP) condition, so that a slight variation in the DC voltage or current levels, can degrade the performance of the sized analog IC. In addition the area of integrated circuits has become an important factor, therefore is necessary to minimize the area without modifying the performance of the circuit. In this manner, this Thesis proposes to guarantee the desired DCOP and minimize the silicon area by applying PSO, MOL and NSGA-II algorithms.

1.5. Objectives

1.5.1. General objective

The main objective of this Thesis is to guarantee a proper direct current operating point (DCOP) condition of the MOS transistor and to minimize the silicon area by applying Particle Swarm Optimization (PSO), Many Optimizing Liaisons (MOL) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) metaheuristics.

1.5.2. Specific objectives

- Application of metaheuristics algorithms to automate the biasing and sizing of analog ICs.
- Apply mono-objective and multi-objective algorithms for the sizing of OTAs using constraints.
- Guaranteeing the DCOP of each MOS transistor to ensure robustness.
- Minimize the silicon area of an OTA.

1.6. Thesis Organization

This thesis is organized in five chapters. In chapter 1 an introduction to the methodology of analog integrated circuit's design, and also the general aspects of the metaheuristics that are used in the sizing of amplifiers are presented. In addition, allusions to the most influential works applied to the sizing of CMOS circuits are introduced.

Chapter 2 presents the topologies of the operational transconductance amplifiers (OTAs) to be optimized through different metaheuristics solving mono and multi-objective problems. In addition, the design of a Two Stage Miller OTA is done by hand using the quadratic model's equations. Finally, the results and configurations used to characterize an OTA are presented.

Chapter 3. In this chapter the theory about mono and multi-objective algorithms, such as PSO, MOL and NSGA-II is described. The principle of operation of each metaheuristic and the modifications performed on the algorithms to carry out the design of the integrated circuits are portrayed. Also, the techniques used to verify the transistor's operation region and a method to estimate the integrated circuit's area considering the minimum dimensions allowed by the technology is explained.

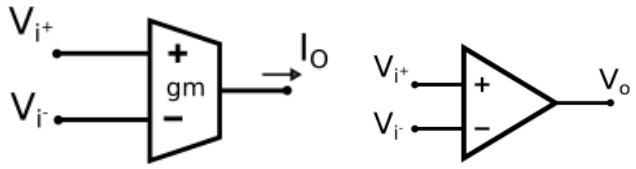
In chapter 4 the results provided by the optimization algorithms are shown and the specifications required to carry out the sizing of the topologies presented in chapter 2 are described. The behavior of the particles corresponding to the PSO and MOL algorithms and the Pareto fronts from NSGA-II are described. The sizing results of the Miller and the RFC-OTA are summarized in tables.

Chapter 5 shows the layout of a Two Stage Miller OTA and the post-layout simulations to verify that the behavior of the circuit is optimal. In addition simulations to test the circuit's performance under PVT variations are performed.

Finally, chapter 6 presents the conclusions of the Thesis and future work.

Chapter 2
DCOP Conditions of Analog ICs

Operational amplifiers (op-amps) are one of the most important integrated CMOS circuits in the development of VLSI systems, and are studied in basic electronic courses. As mentioned in [4], operational amplifiers are mainly employed to drive capacitive loads, such as transistor's gates or in applications like switched capacitor, track and hold circuits and data converters. In this type of application there is no need for the amplifier to have a low output impedance, therefore op-amps are usually replaced by operational transconductance amplifiers (OTAs) which output impedance is high. The symbol of an op-amp and OTA is shown in Figure 2.1.



(a) OTA symbol. (b) Op-amp symbol.

Figure 2.1: Symbols.

Nowadays, due to the high demand of high-speed electronic devices, many researchers have focused on designing integrated circuits like OTAs to work at both high frequencies and low power. The OTA is the primary block in analog integrated circuit's applications like Gm-C filters and sigma delta A/D converters, since the operation of these circuits depends on the OTA's performance.

There is a great variety of OTA's topologies but according to [4], the typical structure of an OTA is represented in Figure 2.2. The first stage corresponds to a differential amplifier, which provides a differential output voltage that depends only on the differential input. Subsequently, the differential signal is converted to single-ended, but this stage can be omitted depending on the application. To guarantee high speed and accuracy, an OTA with high gain and wide bandwidth is the best solution if the gain of the first stage

is not enough, a second stage can be added to improve the OTA's gain. For the implementation of an op-amp an output stage that provides low impedance and improves the slew rate is added. However, if the application does not require a low output impedance, both the OTA's input transconductor and large output resistance can be used to achieve the required voltage gain .

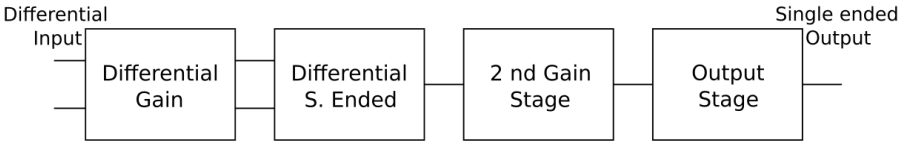


Figure 2.2: Diagram of a typical OTA [4]

In general, the gain of a single-stage OTA is approximately 30 to 40dB, which means that to ensure a gain greater than 60dB is necessary to add two stages or alternatively to increase the output resistance by using topologies such as Folded Cascode. The choice of the topology's amount of stages clearly depends on the application. According to the authors in [27], a performance comparison between single stage amplifiers and two stage amplifiers shows that single stage amplifiers are suitable to operate at the high frequency range while two stage amplifiers are appropriate to perform at the mid-frequency range.

In this work, the Two-Stage Miller and Reclyded Folded Cascode OTA's topologies are selected to be optimized through the use of both mono and multi-objective metaheuristics. The main objective is to reduce is the silicon area, while also guaranteeing a proper DCOP conditions.. As mentioned in [6], the main condition for a MOS transistor to work in strong inversion is modeled by (2.1). The operating region of a MOS transistor is an important factor in the performance of OTAs, so it is necessary to ensure an appropriate DCOP.

$$V_{DS} > V_{GS} - V_{TH} \tag{2.1}$$

2.1. Two-Stage Miller Operational Transconductance Amplifier

As specified by its name the Two Stage Miller is the cascade of two stages, where the first stage is composed of a differential amplifier, integrated by M_1 and M_2 transistors, and a single ended converter formed by M_4 and M_5 transistors, as shown in 2.3. The second stage is a simple inverter with active load, to improve the differential mode gain. Owing to the fact that the output signal of the differential pair is current, which comes from M_2 and is mirrored by M_4 and M_5 to be subtracted by M_1. Therefore a

single ended output voltage is the signal contribution of the two currents multiplied by the output resistance, that is the input signal of the second stage. The differential mode gain of a Two Stage OTA at low frequency is given by the product of the gain of each stage, as seen in (2.2).

$$Av = A_1 A_2 = \frac{gm_1 gm_6}{(gds_1 + gds_5)(gds_6 + gds_7)} \tag{2.2}$$

As mentioned in [28], when the first and the second stage are connected without any intermediate components the frequency response could be disrupted, therefore are various pole compensation techniques. The first compensated topology is the two stage OTA with miller compensation, this technique is based on using a miller capacitor Cc between the two corresponding stages, where the feedback capacitor splits the poles. The pole of the first stage goes at low frequency and the one of the second stage is pushed at high frequency, this is called pole splitting, then the dominant pole will decrease and the non dominant pole will increase simultaneously when the gain of the second stage improves. Another technique is to use the nested miller capacitor along with a nulling resistor between the two stages, this compensated topology is called two stage OTA with miller capacitor and nulling resistor, this topolgy is shown in Figure 2.3. According to [4], the resistance R_C value that cancels the zero is the inverse of the transconductance gain of the second stage's transconductance gain.

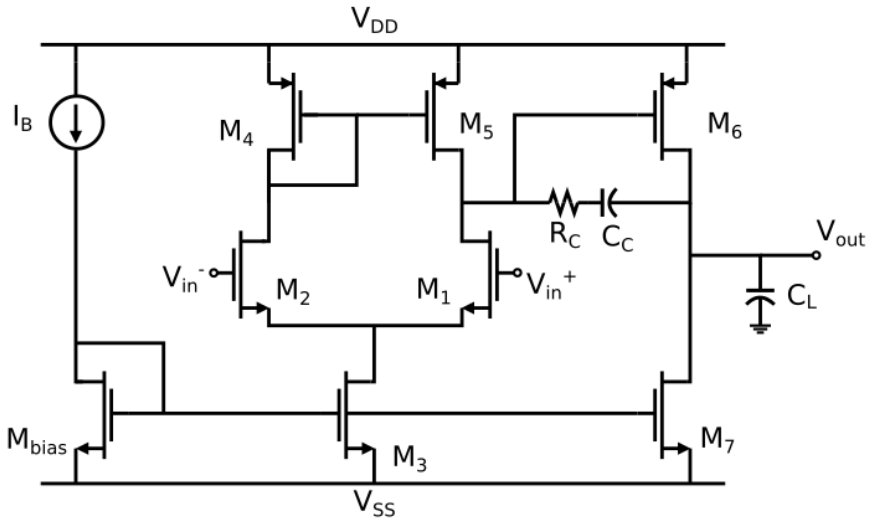


Figure 2.3: Two-Stage Miller OTA.

2.2. Design of a Two-Stage Miller OTA

One of the techniques for the design of analog integrated circuits is to use the quadratic model to carry out the transistor's sizing. Considering the operational transconductance amplifier (OTA) Milller of Figure 2.3 and the specifications shown in Table 2.1.

Table 2.1: Specifications for the design of the Two-Stage Miller OTA

Topology	Miller OTA
CMOS Technology[μm]	0.18
Voltage Supply[V]	$\pm$ 0.9
$(K_n)_n$[$\mu A/V^2$]	340
$(K_n)_p$[$\mu A/V^2$]	70
C_L[pF]	5
GB[MHz]	100
SR[$V/\mu s$]	20
V_{DSAT}[V]	0.1

The equation for the gain bandwidth is given by 2.3.

$$GB = \frac{gm}{2\pi C_L} \tag{2.3}$$

where the transconductance's equation can be approximated to

$$gm = \frac{2I_D}{V_{GS} - V_{TH}} \tag{2.4}$$

Replacing (2.4) in (2.3) is obtained

$$GB = \frac{2I_D}{2\pi(V_{GS} - V_{TH})C_L} \tag{2.5}$$

Then, to find the drain current needed to satisfy the GB value, solve 2.5 for I_D.

$$I_D|_{GB} = \frac{2\pi GB(V_{GS} - V_{TH})C_L}{2} \tag{2.6}$$

Replacing the values of Table 2.1 in (2.6), I_D is

$$I_D|_{GB} = 315\mu A$$

In addition is important to guarantee that the magnitude of the current satisfies the slew rate established in the circuit's specifications. Therefore, to find the minimum value of the current necessary to achieve the required slew rate, the following equation is used:

$$SR = \frac{2I_D}{C_L} \tag{2.7}$$

Equation (2.7) leads to the expression that allows to compute I_D

$$I_D = \frac{C_L SR}{2} \tag{2.8}$$

Then, the minimum current value for I_D is

$$I_D|_{SR} = 50\mu A$$

According to I_D value calculated by 2.8, the minimum value of the current required to have the desired slew rate is 6 times less than the current necessary to reach a $GB = 100MHz$, therefore the minimum current value taken for the following equation is the $I_D|_{GB}$. Using the equation of the quadratic model and solving for W/L, leads to equations 2.9 and 2.10 that correspond to NMOS and PMOS transistors, respectively.

$$(\frac{W}{L})_n = \frac{I_D|_{GB}}{\frac{(K_n)_n}{2}(V_{GS} - V_{TH})^2} \tag{2.9}$$

$$(\frac{W}{L})_p = \frac{I_D|_{GB}}{\frac{(K_n)_p}{2}(V_{GS} - V_{TH})^2} \tag{2.10}$$

According with the specifications, the dimensions rations for both kinds of transistor are

$$(\frac{W}{L})_n = 47$$

$$(\frac{W}{L})_p = 225$$

If $L = 0.18\mu m$, the transistor's width values are

$$(W)_n = 9\mu m$$

$$(W)_p = 41\mu m$$

Finally the values of Rc and Cc are obtained from the following expressions

$$R_C = \frac{1}{2gm} \tag{2.11}$$

Substituting (2.4) in (2.11), 2.12 is obtained

$$R_C = \frac{V_{DSAT}}{4I_D} \tag{2.12}$$

Replacing the values of Table 2.1 in (2.12)

$$R_C = 160\Omega$$

The value of Cc is determined by the following expression

$$C_C \leq \frac{C_L}{5} \tag{2.13}$$

Therefore the compensation capacitor's value is

$$C_C = 1pF$$

The best strategy according to [4], is send the zero at infinite and not trying to enlarge the bandwidth by some pole-zero cancellation.

2.3. Characterization

The characterization of an OTA is an important process, since it allows to verify the circuit's performance. Currently, the behavior of this type of integrated circuits is very close to the behavior of an ideal device, however in VLSI systems precise models are required. Therefore to verify that the performance of the device matches the calculated results, analysis in the DC, AC and time domains are generally carries out.

One of the most important features of OTAs is the differential gain (A_{OL}), which is the open loop differential gain measured as a function of frequency and determines the accuracy of the feedback system. The magnitude of the gain can vary a few orders of magnitude depending on the application, and a high gain is usually required to suppress non-linearity, therefore a typical value of the differential mode gain, ranges from 60 to 80 dB. Fig 2.4 shows the circuit configuration implemented to measure the A_{OL}, GBW and PM. Also, Figure 2.6 shows the measurements results of the differential gain, GBW and phase margin obtained from the design of the Two Stage Miller OTA designed in the previous section.

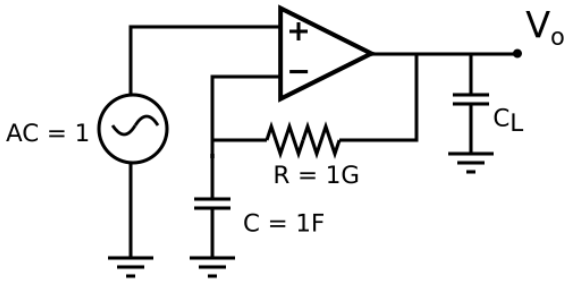


Figure 2.4: Configuration to measure the A_{OL}, GBW and PM.

The gain-bandwidth product GBW (or unity gain frequency (f_T)) is the frequency at which the differential gain is equal to 0dB, as shown in Figure 2.5. The unity gain

frequency is the product of the differential gain (A_{OL}) and bandwidth (f_1), therefore f_T is called gain-bandwidth product.

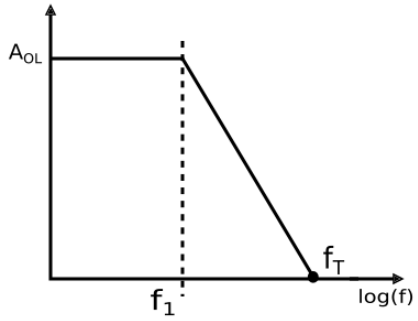


Figure 2.5: Bode Diagram

As mentioned in [4], the phase margin is the phase shift of the small-signal differential gain measured at the unity gain frequency. In order to ensure stability when using the unity gain configuration is necessary to achieve a phase margin greater than 60º.

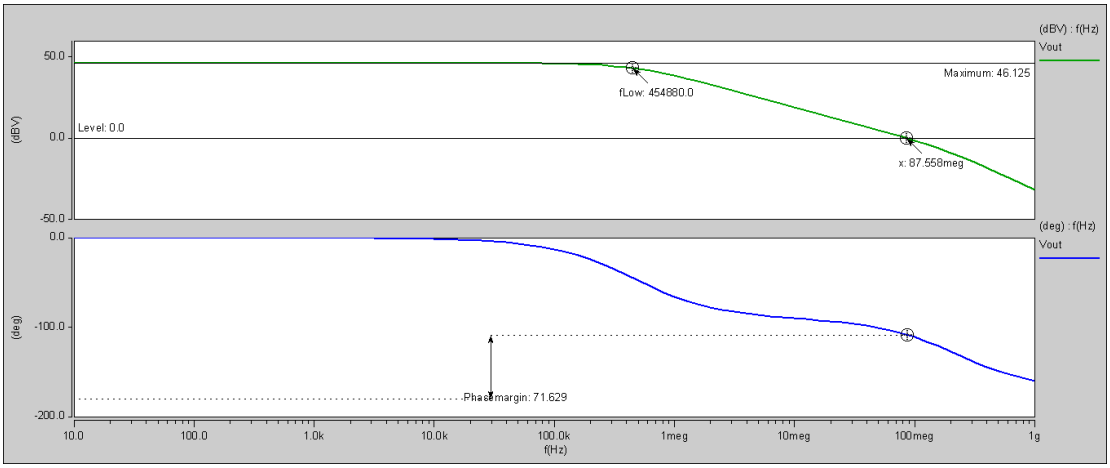


Figure 2.6: Results of the A_{OL}, GBW and PM.

$$A_{OL} = 46dB\ GBW = 87MHz\ PM = 71^{o}$$

Common mode gain (A_{CM}) is the open loop gain obtained by applying a small signal to both inputs. To measure it, the configuration in Figure 2.7 is used. Ideally, OTAs should amplify only differential signals and attenuate common mode signals. A typical value of A_{CM} at low frequency is 10-30dB. The result of the A_{CM} is shown in Fig. 2.8.

$$A_{CM} = -1.5dB$$

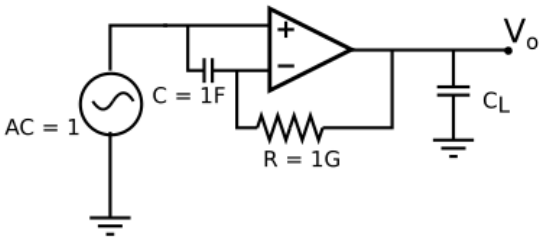


Figure 2.7: Configuration to measure the A_{CM}.

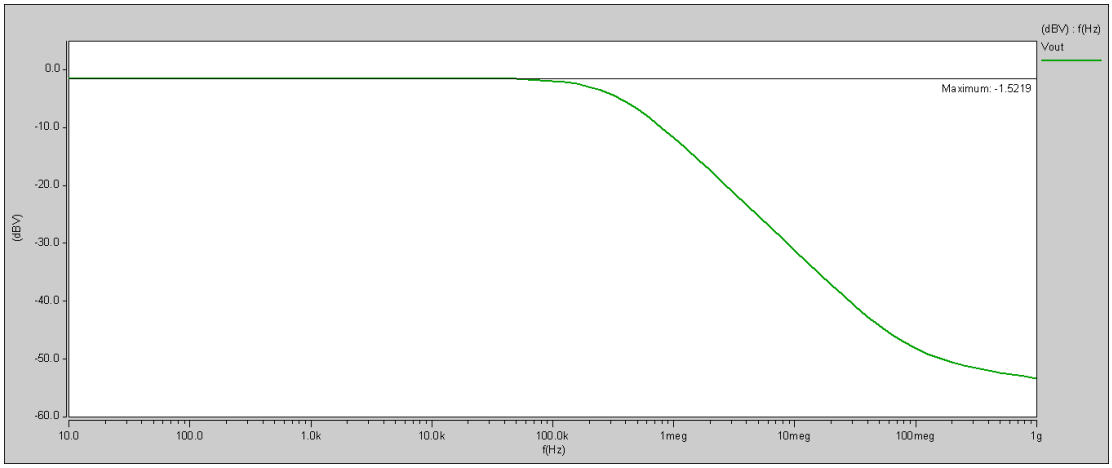


Figure 2.8: Result of the A_{CM}.

A merit factor of any OTA is the common mode rejection ratio ($CMRR$), this measure is the ratio between the differential gain and the common mode gain. The effect of a finite $CMRR$ on an OTA is the presence of an undue signal at the output of the amplifier that is a function of the common mode signal component at its input. Generally a high value of $CMRR$ is desired.

$$CMRR = \frac{A_{OL}}{A_{CM}}$$

The CMRR of the Miller OTA is illustrated in Fig 2.9.

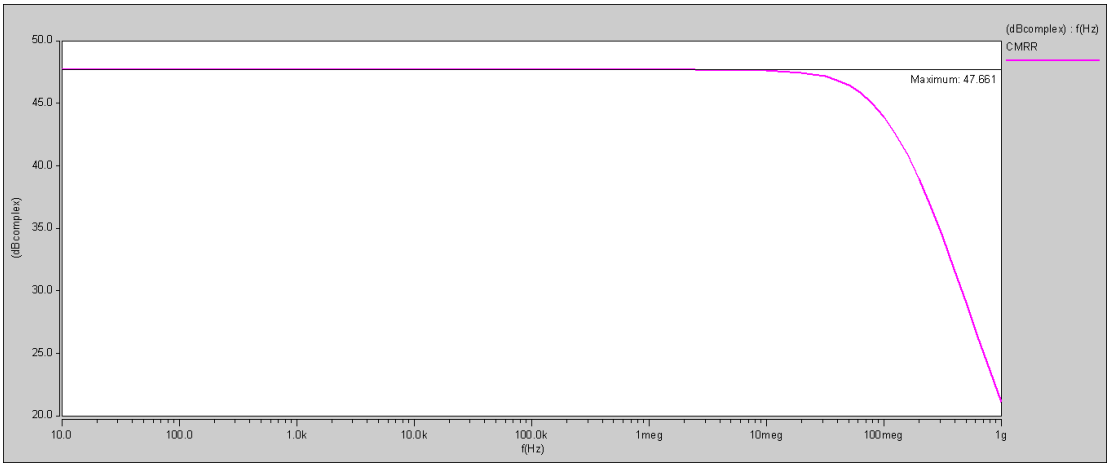


Figure 2.9: Result of the $CMRR$.

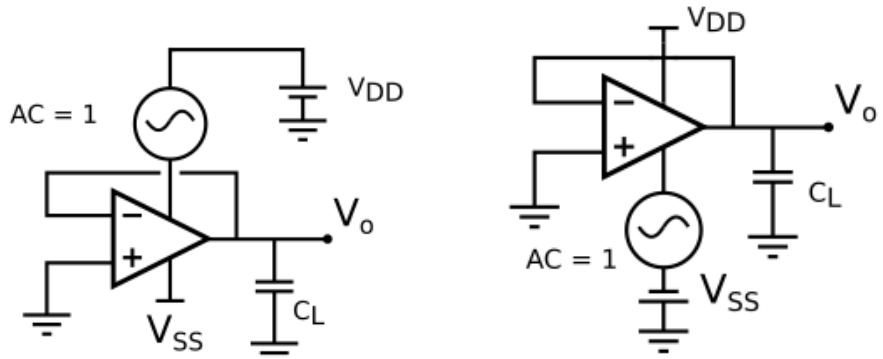
$$CMRR = \frac{46dB}{-1.5dB} = 47dB$$

Another important merit factor for an OTA is the power supply rejection ratio ($PSRR$). This measure is obtained applying a small signal in series with the positive or negative voltage supply. The ration between the differential gain and the power supply gain leads the $PSRR$. Fig. 2.10 illustrate the configuration to measure the $PSRR$. A typical value of $PSRR$ is 60 dB at low frequencies and decreases to 20 – 40 dB at high frequencies.

The output voltage of the Figure 2.10 is equivalent to $Vo = 1/PSRR$, therefore to know the value of the PSRR is necessary to obtain the inverse of the output voltage. Figure 2.11 shows the results obtained from both $PSRR+$ and $PSRR-$.

$$PSRR+ = 51dB \; PSRR- = 54dB$$

The slew rate (SR) is a transient analysis and is defined as the maximum achievable time derivative of the output voltage. The positive slew rate can be different from the negative slew rate, depending on the specific design. Typical SR's values, , fluctuate



(a) $PSRR^{+}$. (b) $PSRR^{-}$.

Figure 2.10: Configuration to measure the $PSRR$.

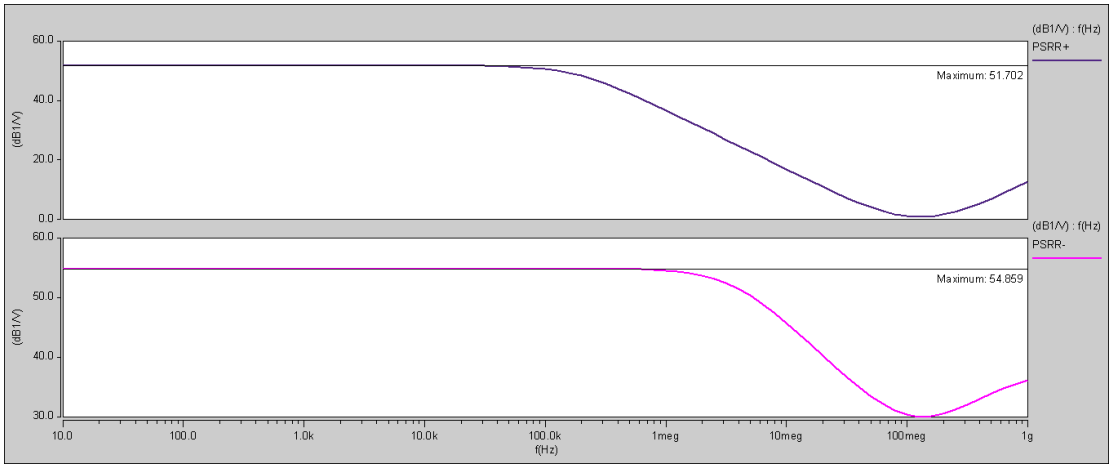


Figure 2.11: Result of the $PSRR+$ and $PSRR-$.

within the 40 and 80$V/\mu s$ range. The configuration to measure SR is shown in Figure 2.12. In addition another important parameter is the settling time, this is the time that the output voltage requires to achieve the expected output voltage. The SR is given by

$$SR = \frac{\triangle V}{\triangle t}$$

According to [29], the applied input signal to obtain a correct measurement of SR in 180nm technology must have an amplitude of 50mVpp at a frequency of 5MHz.

$$SR+ = 7v/\mu s \; SR- = 7v/\mu s$$

Another important figure of merit (FoM) of an OTA is obtained from (2.14) and is typically measured in $((MHz)(pF))/mA$.

$$FoM = \frac{(GBW)(C_L)}{I_B} \tag{2.14}$$

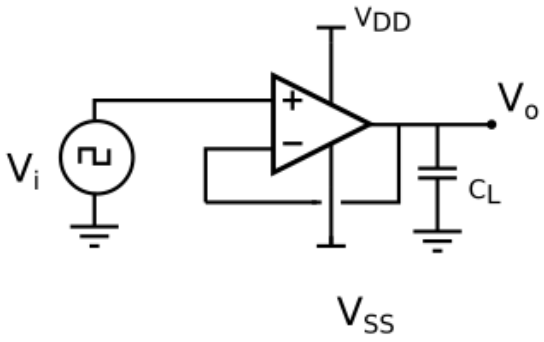


Figure 2.12: Configuration to measure the SR.

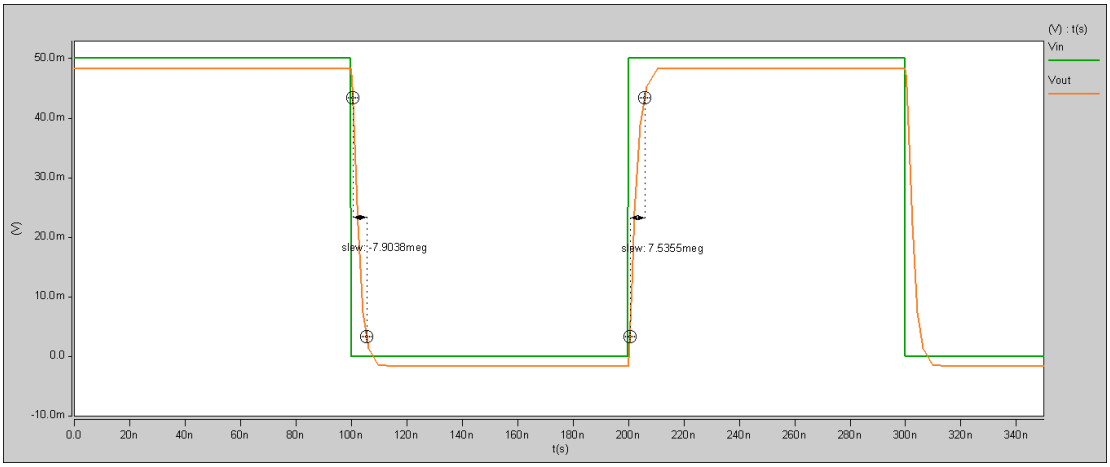


Figure 2.13: Result of the $SR+$ and $SR-$.

Therefore the FoM of the Two Stage Miller OTA designed in the previous section is

$$FoM = 870$$

Finally, the maximum and minimum input voltages are measured using the configuration shown in Figure 2.14. According to [30], to obtain this measurement is necessary to excite the circuit with a triangular signal at the input with a maximum amplitude determined by the power supply voltages. The result is shown in Fig. 2.15

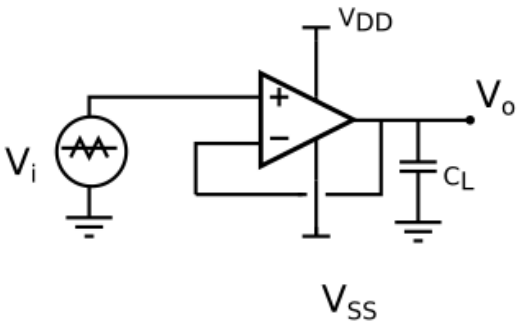


Figure 2.14: Configuration to measure the $Vin+$ and $Vin-$.

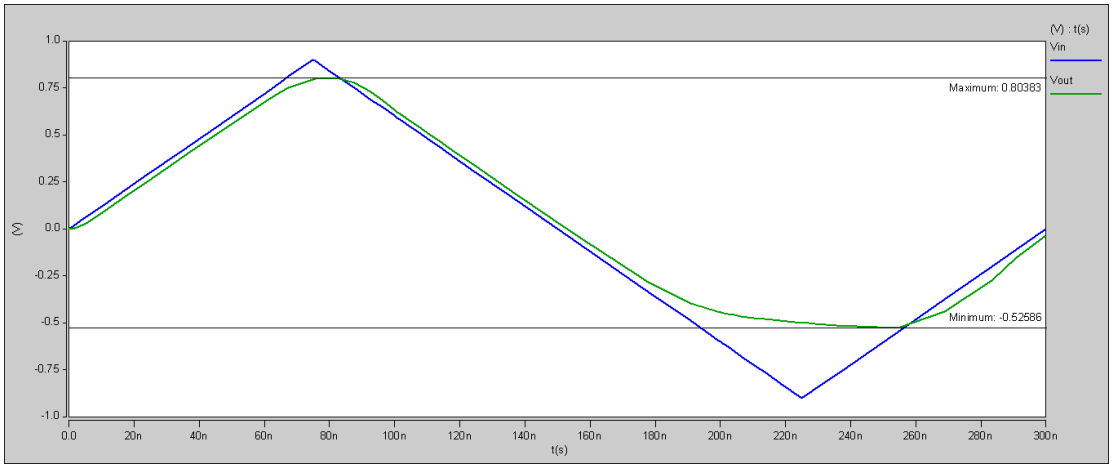


Figure 2.15: Result of the $Vin+$ and $Vin-$.

$$Vin+ = 0.8v \; Vin- = -0.5v$$

2.4. Recycled Folded Cascode Operational Transconductance Amplifier

The RFC OTA shown in Fig. 2.16 is a modification of the traditional Folded Cascode OTA, it converts the folding node current source of FC OTA into a driven current source

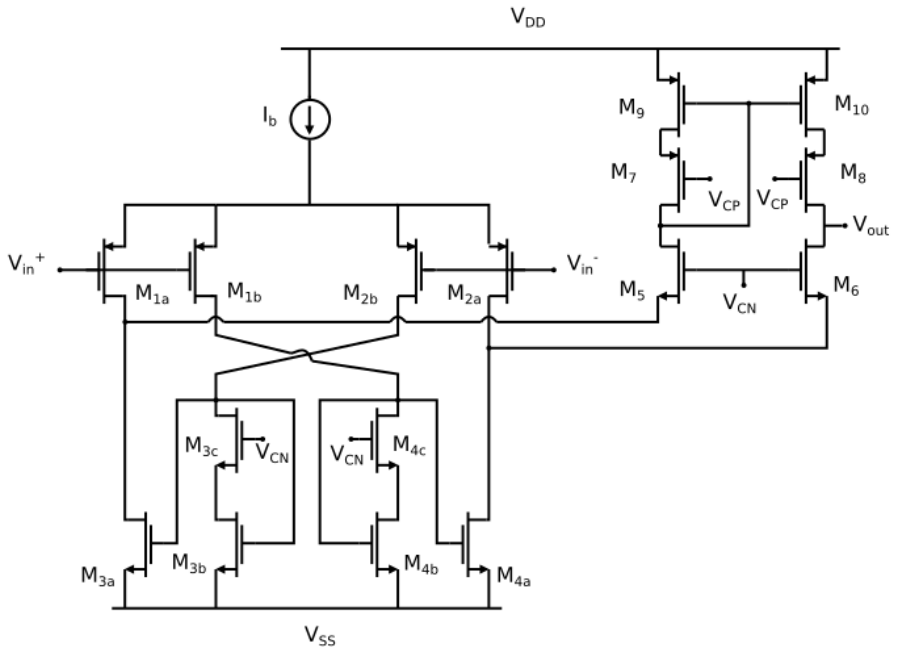


Figure 2.16: Reclycled Folded Cascode OTA.

by using a recycling current mirror with a gain equal to K. The RFC OTA improves the transconductance and the slew rate of the FC OTA. The differential pair of the FC OTA is divided in half to fix the current to $I/2$, which results in $gm = 2gm_{a1}$. Another improvement is achieved by using a cross coupled format at the differential pair's output with the objective of increasing the bandwidth.

One of the disadvantages of the RFC OTA is its poor power efficiency, this is mainly due to two reasons as mentioned in [31]. Firstly, current gain K scales the dynamic and static current. Secondly, the active current mirrors included, lead to an internal copy of dynamic currents. According to [29], by applying good approximations, the main relations of the RFC OTA are summarized through equations (2.15-2.20). The transconductance Gm_{RFC} is obtained

$$Gm_{RFC} = gm_{a1}(1 + k) \tag{2.15}$$

For the power and area to be equal to that of the FC OTA, K must be equal to 3. If $K \neq 3$, transistors M9 to M14 need to be scaled accordingly, in order to maintain the same inversion level. It has been demonstrated that the transconductance of the RFC OTA is twice than that of the FC OTA with the same power consumption. Therefore the RFC OTA has twice the gain bandwidth product (GBW) as that of the FC OTA for the same power, and consequently twice the speed. In addition the DC gain Av_{RFC} is

$$Av_{RFC} = R_{OUT}Gm_{RFC} = R_{OUT}gm_{a1}(1 + k) \tag{2.16}$$

where

$$R_{OUT} = (gm_{12}r_{o12}r_{o14})||gm_{10}r_{o10}(r_{oa4}||r_{oa2}) \tag{2.17}$$

The gain enhancement depends on the r_{ds} increase of M_{2a} and M_{4a}, since it drives less current compared to their counterparts of the FC OTA. Typically the gain in a RFC OTA is approximately 8 to 10dB greater, in regards to a FC OTA. Another critical design aspect is the Slew Rate (SR), according to [32], by assuming a capacitive load and a large signal in the inputs of the RFC OTA the SR can be derived as follows: if Vin^+ increase M_{1a} and M_{1b} turn off, which force M_{4a} and M_{4b} to turn off. Consequently, the drain voltage of M_{4a} rises and M_{10} is turned off, whereas M_{2a} is driven into deep triode. The tail current, $2I_B$, is driven into M_{2b} and is mirrored by a K factor into M_9 and again by a factor of 1 into C_L. This result's in

$$SR_{RFC} = \frac{2KI_B}{C_L} \tag{2.18}$$

The frequency response of the RFC OTA is defined by both the dominant pole frequency which is determined by the output impedance and the capacitive load,

$$wp_{RFC} = -\frac{1}{R_{OUT}C_L}, C_L = C_{OUT} + C_{d12} + C_{d10} \tag{2.19}$$

and by the unity gain frequency

$$wu_{RFC} = \frac{Gm_{RFC}}{C_L} = \frac{(K+1)gm_{a1}}{C_L} \tag{2.20}$$

Chapter 3
Metaheuristics

The metaheuristics used in the design of integrated circuits have presented an important development, due to the positive results on integrated circuit's optimization. Although of the important advances in this area, none of the works in the state of the art are focused on guaranteeing that the transistor operates in the desired region. As shown in Chapter 2, assuring that a transistor's operation point is set at the desired region is fundamental in the integrated circuit's performance. In this manner, PSO, MOL and NSGA-II algorithms are applied to minimize and maximize the most important features in the design of integrated circuits, such as operational transconductance amplifiers, while guaranteeing that all MOS transistors work in strong inversion. In addition, the step-by-step procedure that follows each of the algorithms for the design of integrated circuits and how the electrical characteristics are evaluated through linking it with a circuit simulator such as SPICE is explained.

3.1. Mono-Objective Optimization

3.1.1. Particle Swarm Optimization

Particle Swarm Optimization belongs to the field of Swarm Intelligence and Collective Intelligence and is a sub-field of Computational Intelligence [33]. Moreover is related to other swarm intelligence algorithms such as Ant Colony Optimization [13]. PSO is based on a mathematical model developed by Kennedy and Eberhart in 1995 [5]. It describes the social behavior of birds or fishes. The model is based on the basic principles of self-organization that are used to describe complex systems. These swarms form a search group with the purpose of finding food, usually each individual continues his search according to his own experience and the experience of the group.

The main objective of the PSO algorithm is that all the particles locate the optimal solutions in a search space. For this is important to initialize both the random positions of all particles in space and the small initial random velocities. In addition, the development of PSO is related to two researches: Evolutionary algorithms and Artificial life. As de-

tailed in [34], PSO initializtion starts from a set of randomly distributed particles in a limited search space, these particles have an initial position and velocity that are represented by simple mathematical expressions. These expressions suggest the movement of each particle towards the best position as an individual and the best global position, besides there are different variants using different update rules. The general idea is to initialize a set of particles in a search space, this gives the particles a favorable initial position, in addition an initial velocity vector is assigned, which allows the particles to change their position in each iteration while the speed is adjusted depending on some random parameters. Each particle can remember its best position and recognize if its current position is the best among the other particles, that is, the best global. The particles have to be updated according to their positions and past speeds. Mathematically, the updating equations are given in (3.1) and (3.2)

$$v_i(t+1) = v_i(t) + c_1\text{rand}()(p_{best}(t) - p_i(t)) + c_2\text{rand}()(g_{best}(t) - p_i(t)) \tag{3.1}$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \tag{3.2}$$

where $v_i(t+1)$ and $p_i(t+1)$ represent the velocity and position of the particle in the i_{th} iteration, respectively; where rand() is a function that returns random real number values between 0 and 1; p_{best} and g_{best} represent the best position of the particle and the best global position among all the particles; c_1 and c_2 are two parameters that represent the confidence of the particle itself (cognition) and in the swarm (social behavior), respectively. These last constants are the most relevant in (3.1), according to a set of tests it was found that the higher the constants the faster the convergence will be. As mentioned in [5], the constants c_1 and c_2 have values that may improve the convergence, but it depends on the kind of problem. In this article $c_1 = c_2 = 2$. In Fig 3.1 is shown the flowchart of the PSO algorithm.

3.1.2. Many Optimizing Liaisons

MOL is a variant of PSO that is based on eliminating the best known position of the particle (p_{best}) in (3.1), which updates to (3.3). This variant behaves similarly or better than the PSO algorithm, where it proposes several combinations for the MOL parameters w, c_2, given in (3.3), and the number of individuals (particles) for different problems, and thus one calibrates the algorithm depending on the dimensions of the problem and the number of evaluations of the objective function. MOL is a purely social algorithm tending to follow the best swarm's particle (g_{best}), thus when the inertia coefficient $w = 1$, it restricts the particles exploring better solutions in the search space, so that a challenge is finding the appropriate w value that allow the particles exploring different directions to find better solutions in the entire search space, in addition to maintain the velocity's limits in a previously defined range. The flowchart of the MOL algorithm is shown in Fig. 3.2.

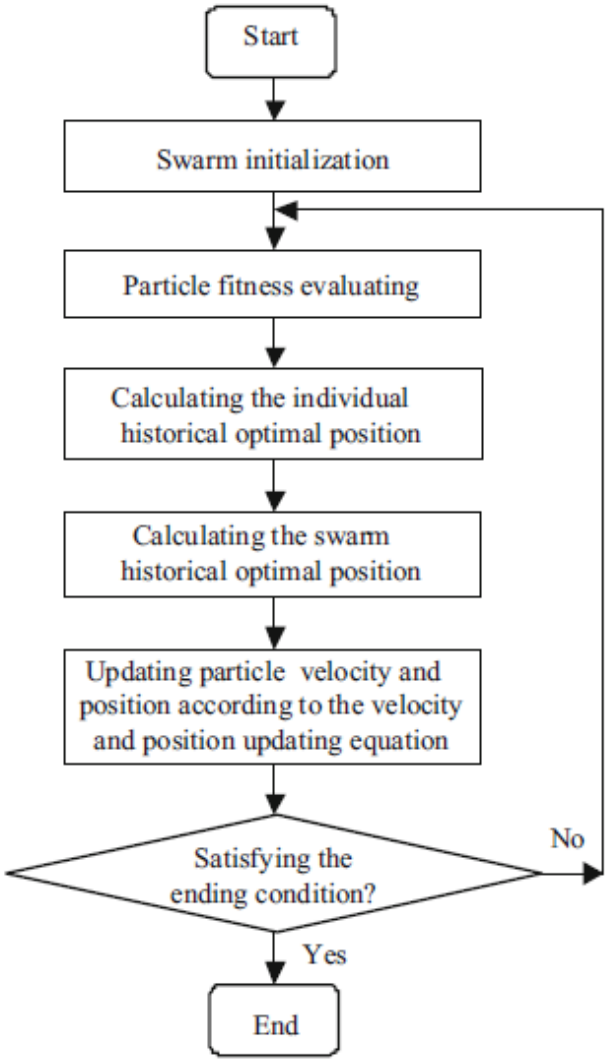


Figure 3.1: Flowchart of the PSO algorithm [5].

$$v_i(t+1) = wv_i(t) + c_2\text{rand}()(g_{best}(t) - p_i(t)) \tag{3.3}$$

One of the main problems with the MOL algorithm is that the parameters w and c_2 are not currently well defined, therefore a challenge is finding the appropriate w value that allow the particles exploring different directions to find better solutions in the entire search space, in addition to maintain the velocity's limits in a previously defined range. In [35] the values of the parameters for several problem configurations are presented, for the case of study of this work the best value for each parameter, considering that is a problem with four design variables and a thousand fitness evaluation, $w = -0.3084$ and $c_2 = 2.0273$.

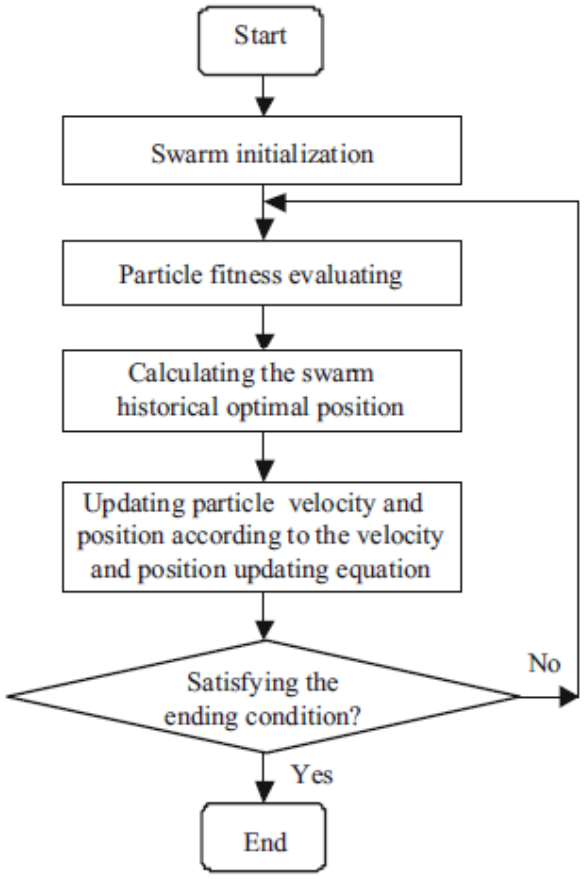


Figure 3.2: Flowchart of the MOL algorithm.

3.2. Multi-Objective Optimization

As explained in [36] a multi-objective problem (MOP) is defined as the problem of finding the vector: $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ that complies with the k inequality constraints:

$g_i(\mathbf{x}) < 0$; for $i = 1, 2, ..., k$, the p equality constraints: $h_j(\mathbf{x}) = 0$; for $j = 1, 2, ..., p$ and minimizes the vector function

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x})]^T$$

In MOP there are many feasible solutions but in order to say that a solution dominates another one, the dominating solution needs to be better in at least one objective and not worse in any objective. This is called Pareto dominance and is defined as follows: A vector $\mathbf{u} \in \mathbb{R}^m$ dominate a vector $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{u} \prec \mathbf{v}$, if and only if $\mathbf{u}$ is less than $\mathbf{v}$.

3.2.1. Evolutionary Algorithms

As described in [33] Evolutionary Algorithms(EAs) belong to the Evolutionary Computation field and are inspired by the process and mechanisms of biological evolution and the natural selection of the species theory presented by Darwin. One of the most important evolutionary algorithms are the genetic algorithms (GA), which were developed by John Holland and his collaborators between 1960 and 1970. According to [11] the base of GA are the genetic operators such as crossover, mutation and selection.

The procedure of a genetic algorithm involves the encoding of an optimization problem from a bit or character strings to represent the chromosomes, the operations of strings by genetic operators, and the selection of the fitness with the aim to find a solution to the problem. Then the genetic operators are explained.

- Crossover is the genetic operator with the higher probability, also as mentioned in [20] is responsible for the cutting and recombination of building blocks. The simplest form of crossover is that a single point is chosen on two chromosomes of equal length and intersecting at that particular point as shown in Fig. 3.3
  In addition is possible to select two or more crossover points but this could degrade the performance. Basically crossover consists of generating a new solution from parameters that are taken from one solution and exchanged with another at the same point.

- The genetic operator mutation is defined as a genetic manipulation operator, this process is random and has a small probability. Essentially is based on altering the gene of a chromosome from one generation to the next. It is usually used to avoid a premature convergence that can occur in the selection process since the solutions become similar before reaching an optimal solution. Figure 3.4 illustrates the operation of the mutation.

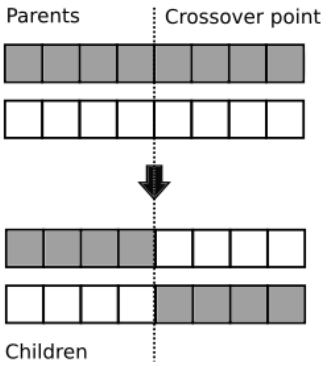


Figure 3.3: Crossover.

Figure 3.4: Mutation.

- Selection is the most important genetic operator and is carried out by the evaluation of its fitness. According to [36] there are three different ways to choose individuals namely Tournament Selection, Fitness Proportional Selection and Rank Based Selection. The former two or more individuals are randomly selected from the current generation to compete with each other. The winner is the individual with the highest fitness. In proportional selection all individuals have a chance of selection but the individual with the highest fitness has more possibilities. Roulette Wheel Selection is the most used method. Finally, the rank of the individual within the generation is used to select the best individual. The elitism is when the fittest individuals in a generation are cloned into the next generation in order to make sure to preserve their genetic material.

3.2.2. Non-dominated Sorting Genetic Algorithm II

The Non-Dominated Sorting Genetic Algorithm II is a non domination multi-objective optimization that was proposed by Kalyanmoy Deb in [37]. In NSGA-II the solution is obtained from the current parents and their offspring using a fitness function that is defined as a measure of how good is in relation to the objective and constraints.

In the NSGA-II algorithm all the non-dominated solutions form the Pareto front, which is approximated by sorting and ranking all solutions in order to choose the better solutions to make new offspring. This algorithm belongs to evolutionary algorithms which

use a set of solutions also called population, which is evolving through genetic operators to generate solutions that represent individuals, only the best individual survive to the next generation. According to [36] NSGA-II is based on three main approaches to improve the performance of the algorithm: Fast Ranking Function, Crowding Distance Assignment and Elitism. In Fig. 3.5 is shown the Pareto front and the crowding distance.

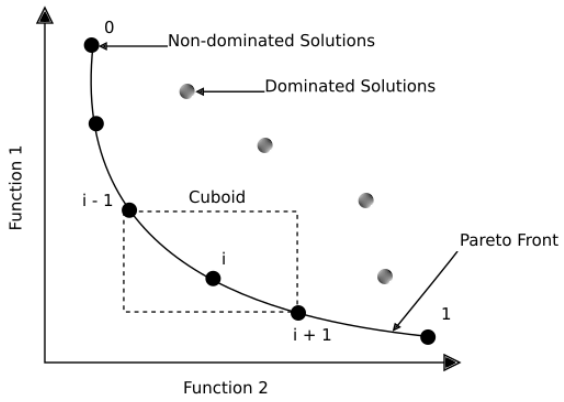


Figure 3.5: Pareto Front and Crowding Distance.

Fast Ranking Function determines a rank value according to the number of individuals that dominated each solution, as mentioned in [38] the non-dominated solution receive a rank equal to one, while the other solutions receive a rank value according to how many subsets they dominate. Crowding Distance measures the average size of the cuboid formed with the points that enclose a solution in the population. Finally Elitism basically select the individual that best reaches the fitness function and the constraints, in other words the individual with the highest rank. If all the solutions are in the first rank the algorithm select the solution with the highest crowding distance. In Fig. 3.6 is illustrated the flowchart of the NSGA-II algorithm.

Is important that the Pareto front has a good convergence, in addition, the solutions must be spread all over the front, in figure 3.7 different Pareto fronts are shown. In quadrants 3 and 4 is seen how the solutions are not well distributed along the front, in quadrant 2 the front is not well defined since this front does not have feasible solutions. Finally, in quadrant 1, the ideal Pareto front is seen.

3.3. Sizing ICs by PSO, MOL and NSGA-II

Analog integrated circuit's design is a complex process due to the many targets and trade-offs that are involved in the design process, for that reason optimization methods as metaheuristics are used for transistor sizing. The algorithms used for the design of

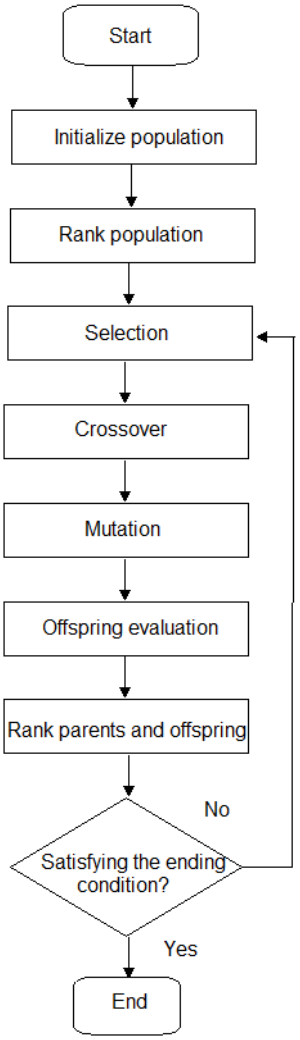


Figure 3.6: Flowchart of the NSGA-II algorithm.

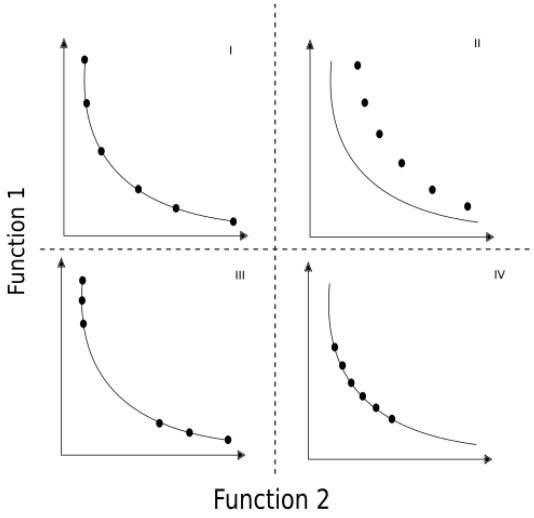


Figure 3.7: Pareto Fronts.

integrated circuits are applied especially to analog circuits where the design variables are the relation W/L and the biasing conditions. Generally, algorithms are developed to optimize problems that can be interpreted through mathematical functions, hence the challenge for the optimization in the design of analog integrated circuits is to know how to interpret the circuit as a function, therefore is important to make an analogy between the devices that are part of the circuit and the variables to be introduced to the algorithm.

3.3.1. Codification

In order to make an analogy is necessary to know the terminology used in metaheuristics based on swarm optimization and evolutionary algorithms. Some of these are: fitness function, particle or individual, position or gene, velocity, swarm (population) and iteration also called generation. The design variables represents the gene of the individual in a search space, the velocity of each particle in the case of PSO and MOL algorithms reflects the distance traveled by this particle at each iteration and a chromosome represents a vector formed by genes. In Fig. 3.8 the coding of a simple current mirror is illustrated.

A population is formed by a set of individuals, an objective function and even constraints. Finally the generation is defined as iteration and correspond to the evolution process and each design variable has a search space specified by the designer and its limits are established so that the solutions converge faster and avoid errors with the integrated circuit simulator.

The design variables W, L and I_B are encoded by integer numbers and scaled in microns into SPICE using the command .option scale by $0.09\mu m$ which is equivalent to

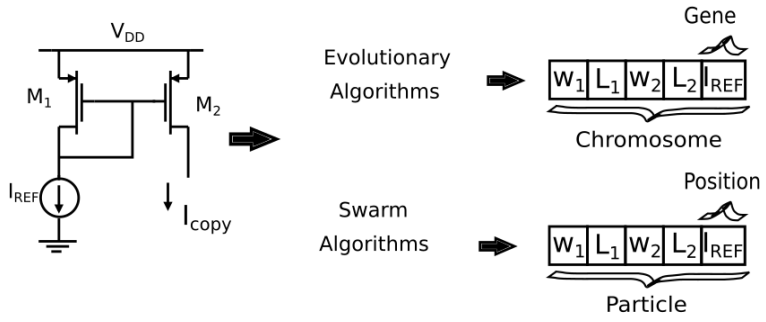


Figure 3.8: Circuit Codification.

the lambda of the $180nm$ CMOS technology. Usually, the values of the design variables are written as shown in the following SPICE code:

.param W1 = 9u
.param W2 = 54u
.param L1 = 0.18u
.param L2 = 0.36u

The command .param is used to parameterize the design variables, moreover an example of using .option scale command within SPICE is given below:

.option scale = 0.09u
.param W1 = 100
.param W2 = 600
.param L1 = 2
.param L2 = 4

Mx Drain Gate Source Bulk PMODEL W ='W1' L ='L1'
Mx Drain Gate Source Bulk NMODEL W ='W2' L ='L2'

The post-processing step for rounding the sizes of W and L allows to decrease the computation time and memory usage, it also facilitates the coding of the design variable's values. Another advantage of scaling by values that are multiples of lambda is that it facilitates the realization of the layout.

3.3.2. Initialization

PSO and MOL algorithm implementation was developed in MATLAB^{TM} by [39] and NSGA-II by K. Deb. The evaluation of the fitness function is carried out by linking the circuit simulator (SPICE) with the algorithms. The first step is to generate an input file of the integrated circuit (netlist) in a SPICE file (.sp). The design variables of

the integrated circuit, the number of individuals, generations, objective function, constraints, stop criteria and search space of each design variable are declared previously as independent variables into the algorithm.

The algorithm randomly initializes the circuit design variables within a given search space, these values are assigned to the design variables w_1, $w_2...w_n$, Ib and subsequently replaced in the netlist using the command .param

.param W1 = W2 = W3 = Ib =

3.3.3. Fitness Evaluation

After coding the design variables and initializing the parameters corresponding to the metaheuristics algorithm the input file is simulated using SPICE. According to (3.1) and (3.3) the position of the particle that corresponds to the design variables of the integrated circuit can give negative solutions, this represents a problem in the numerical method used by SPICE, therefore is important to establish the limits of the search spaces, considering that the lower limit for transistor's width W needs to be greater than the upper limit of the transistor's length L. This to avoid problems with the circuit simulator, since it is physically impossible that W has a value less than L.

After simulating the circuit is necessary to measure the performance features that corresponds to the objectives and constraints of the analog IC. To perform the analysis in different domains (.DC, .AC, .TRAN) subcircuits are used in different configurations. In order to obtain the characteristics of the integrated circuit that correspond to the values of the objectives and constraints, the .MEASURE command is used, which gives the numerical outputs results in the SPICE .lis output file. Therefore the metaheuristic algorithm is modified to capture this data through a search within the .lis file. These values are stored within the metaheuristic algorithm to be later used to determine the best particle and best global particle. The format to use this command is as follows:

```
.OP
.AC dec 10 10 1000meg
.TRAN 0.01n 600n
.MEASURE AC DCgain MAX Vdb(Vouta)
.MEASURE AC GBW when Vdb(Vouta)=0
.MEASURE AC CMgain MAX Vdb(Voutb)
.MEASURE AC PSRRp MIN Vdb(Voutc)
.MEASURE TRAN Vmax MAX V(Voutf)
.MEASURE TRAN Vmin MIN V(Voutf)
```

The SPICE output file (.lis) shows the results obtained through each of the analyzes.

These values are read by the metaheuristic algorithm, and immediately assigned to the variables that represent the objectives and constraints to perform the design process. The format in .lis output file is as follows:

dcgain= 7.1062E+01 at= 1.0000E+01
from= 1.0000E+01 to= 1.0000E+09
gbw= 1.3756E+07
cmgain= -1.3915E+01 at= 1.0000E+01
from= 1.0000E+01 to= 1.0000E+09
psrrp= -7.6328E+01 at= 1.0000E+01
from= 1.0000E+01 to= 1.0000E+09
psrrn= -7.4728E+01 at= 5.0119E+01
from= 1.0000E+01 to= 1.0000E+09

3.3.4. DCOP Conditions

Both sizing algorithms guarantee DCOP conditions by verifying that all MOS transistor operate in strong inversion, as mentioned in Chapter 2, specifically in (2.1). In this manner, to guarantee that the MOS transistors are working in strong inversion, is sufficient to accomplish $3XSAT$, where SAT is evaluated by (3.4).

$$SAT = \frac{V_{DS}}{V_{GS} - V_{TH}} \tag{3.4}$$

The DCOP condition is guaranteed by extracting the values of V_{DS}, V_{GS} and V_{TH} for each MOS transistor and from the output text-file provided by SPICE, with extension .lis. An example of this text file is shown in Table 3.1.

Each column is verified to accomplish that the ratio in (3.4) is greater than 3($V_{GS} - V_{TH}$), so that the DCOP of the MOS transistor is in strong inversion, as shown in Figure 3.9. If the ratio is lower than 3, the MOS transistor is in moderate inversion and may lead to triode region, which is not appropriate for the OTAs. If all MOS transistors are in strong inversion, PSO, MOL and NSGA-II algorithms evaluate the MOS area of the OTA, and update the best individual associated to the lowest MOS area of the population.

Within SPICE the values of the voltages V_{DS}, V_{GS} and V_{TH} for each transistor are selected to perform the operation corresponding to (3.9) using the command .MEASURE. The format for this command to perform the operation is as follows:

.MEASURE TRAN SAT1 MAX PAR('vds(X1.M1)/((vgs(X1.M1)-vth(X1.M1)))')
.MEASURE TRAN SAT2 MAX PAR('vds(X1.M2)/((vgs(X1.M2)-vth(X1.M2)))')

Table 3.1: Output file .lis provided by SPICE.

element	1:m1	1:m2	1:m3
model	0:n_18_mm	0:n_18_mm	0:n_18_mm
region	Saturati	Saturati	Saturati
id	23.2156u	23.0051u	46.2207u
ibs	-266.9393a	-266.9393a	-7.91E-21
ibd	-888.7196a	-937.5297a	-176.8286a
vgs	553.7568m	551.9880m	586.2443m
vds	806.5237m	869.8355m	346.2432m
vbs	-346.2432m	-346.2432m	0
vth	524.2255m	523.0335m	477.5087m

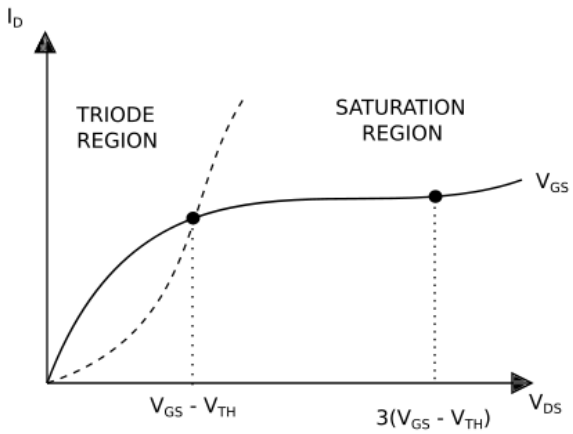


Figure 3.9: DC OP regions in a MOS transistor plotting I_D vs V_{DS}.

.MEASURE TRAN SAT3 MAX PAR('vds(X1.M3)/((vgs(X1.M3)-vth(X1.M3)))')
.MEASURE TRAN SAT4 MAX PAR('vds(X1.M4)/((vgs(X1.M4)-vth(X1.M4)))')
.MEASURE TRAN SAT5 MAX PAR('vds(X1.M5)/((vgs(X1.M5)-vth(X1.M5)))')
.MEASURE TRAN SAT6 MAX PAR('vds(X1.M6)/((vgs(X1.M6)-vth(X1.M6)))')
.MEASURE TRAN SAT7 MAX PAR('vds(X1.M7)/((vgs(X1.M7)-vth(X1.M7)))')
.MEASURE TRAN SAT8 MAX PAR('vds(X1.Mbias)/((vgs(X1.Mbias)-vth(X1.Mbias)))')

After performing the simulation, the results of the operation are shown in the output text file .lis of SPICE and are saved as variables within the metaheuristic algorithm.

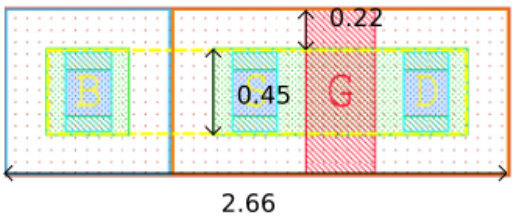
sat1= 6.7225E+00
sat2= 6.9149E+00
sat3= 3.5197E+00

sat4= 6.0894E+00
sat5= 6.2837E+00
sat6= 7.4124E+00
sat7= 5.0344E+00
sat8= 3.9309E+00

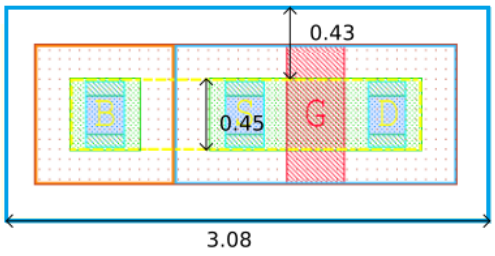
These values are normalized, if the saturation value is greater than 3 then it will be 1 otherwise it will be 0. Finally, the sum of the normalized values of all the transistors is done. If all are equal to 1 then is said that the particle is feasible, but if at least one value is different from 1 the particle is not feasible.

3.3.5. Silicon Area Estimation

The objective function for PSO, MOL and NSGA-II algorithms is the minimization of the silicon area, since allows secondary objectives such as differential mode gain to be in conflict with the objective function, which is a requirement for optimization problems applying metaheuristics. Ideally is not possible to know exactly the total silicon area that an integrated circuit can occupy. However, an estimation can be carry out if the minimum dimensions required to build a transistor, either n-type or p-type, are known, as shown in Figure 3.10.



(a) N-Type.



(b) P-Type.

Figure 3.10: Minimum dimensions of a transistor in 180nm.

Therefore, to obtain an approximation of the total silicon area, the sum of the areas

of each of the transistors that are part of the integrated circuit is performed. Equations (3.5) and (3.6) are used to obtain the area of a n-type and p-type transistor, respectively.

$$N - Transistor_{AREA} = ((w)(0.09\mu m) + 0.44\mu m)(2.66\mu m) \tag{3.5}$$

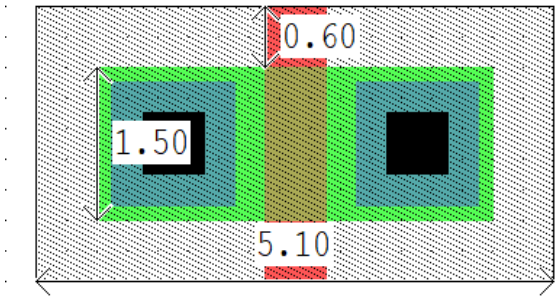
$$P - Transistor_{AREA} = ((w)(0.09\mu m) + 0.86\mu m)(3.08\mu m) \tag{3.6}$$

As observed in the equations (3.5) and (3.6) that determine the area of each transistor, the only variable is the width of the transistor. Where the $0.09\mu m$ factor, serves to scale the value of the W obtained from the algorithm as an integer. Is also considered that $L = 0.36\mu m$, therefore the transistor's area depends only on the change of W. For example, the area of a n-type transistor as shown in Figure 3.10, will be $2.3674\mu m^2$. If the metaheuristics algorithm finds a W value equal to 100, then the value of the transistor's area will be $25.1104\mu m^2$. The corresponding operations are performed within the metaheuristic algorithm.

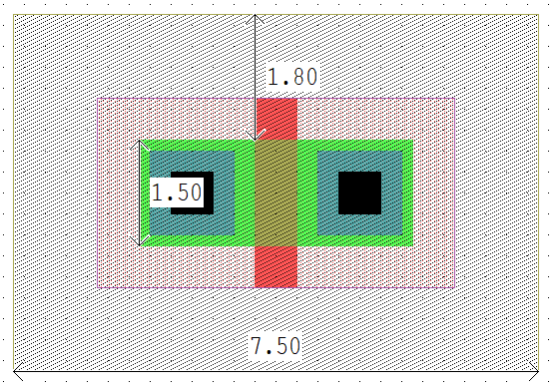
In addition, to perform the RFC OTA optimization the 0.5um onsemiconductor technology is used, where the values of the minimum transistors change. Figure 3.11 shows the values used in equations (3.7) and (3.8) to estimate transistor's area.

$$N - Transistor_{AREA} = ((w)(0.6\mu m) + 1.2\mu m)(5.1\mu m) \tag{3.7}$$

$$P - Transistor_{AREA} = ((w)(0.6\mu m) + 3.6\mu m)(7.5\mu m) \tag{3.8}$$



(a) N-Type.



(b) P-Type.

Figure 3.11: Minimum dimensions of a transistor in 0.5um.

3.3.6. Selection

To carry out the selection of the particle, is necessary to consider the values of the objectives and constraints. Therefore, the best individual is the one that best complies with these values and is selected by means of constraints handling proposed in [40]. This mechanism is based in comparing feasible solutions and the fitness values of the individuals. If two solutions are compared then the individual with the highest fitness value wins, if one of the individuals is infeasible and the other is feasible, the feasible individual wins. When two infeasible individuals are compared the individual who best meets the constraints wins.

Once the simulation is performed, the extraction of the results and the selection of the best individual is passed to the next generation or iteration where the values of the design variables are updated according to the updating equations and genetic operators of the metaheuristics algorithms. Then the following steps are repeated until reaching a stop criteria. PSO, MOL and NSGA-II pseudo-code are summarized in Algorithms 1 and 2 respectively. The only difference between PSO and MOL algorithms in the pseudo-code is when the velocity of the particle is updated, in this case, the pseudo-code of MOL is the same as PSO but replacing (3.1) by (3.3). Is important to mention

that the constraints are normalized between 0 and 1 to facilitate the calculation of the number of constraints that were fulfilled by each particle.

Algorithm 1 PSO

procedure PSO($nPop, MaxIt$)
    Generate the input file of the IC (netlist) according to SPICE
    for $i = 1 : nPop$ **do**
        Initialize randomly the design variables: width (W) of the MOS transistors and the bias current (I_B)
        Simulate the IC$_i$ in SPICE
        Calculate the constraints and update the p_{best} particle
        Update the g_{best} particle by checking the constraints.
    end for
    for $it = 1 : MaxIt$ **do**
        for $i = 1 : nPop$ **do**
            Copy particle i to p
            Update the particle p velocity according to (3.1)
            Update the particle p position according to (3.2)
            Compare particles i and p
            Update p_{best} and g_{best} particles by checking the constraints.
        end for
    end for
end procedure

Algorithm 2 NSGA-II

procedure NSGA-II($N, g, f(x)$)
Generate the input file of the OTA (netlist) according to SPICE
Initialize randomly the design variables: width (W) of the MOS transistor and the bias current (I_B).
Simulate the IC$_i$ in SPICE.
Calculate objective values and constraints
Assign Rank based in Pareto
Generate Child Population
Binary Tournament Selection
Recombination and Mutation
for $i = 1 : g$ **do**
for Each Parent and Child in Population **do**
Assign Rank based on Pareto
Generate sets of nondominated solutions
Determine Crowding Distance
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
end for
Select points (Elitist) on the lower front and are outside a crowding distance
Create next generation
Binary Tournament Selection
end for
end procedure

Chapter 4
Feasible sized solutions provided by PSO, MOL and NSGA-II

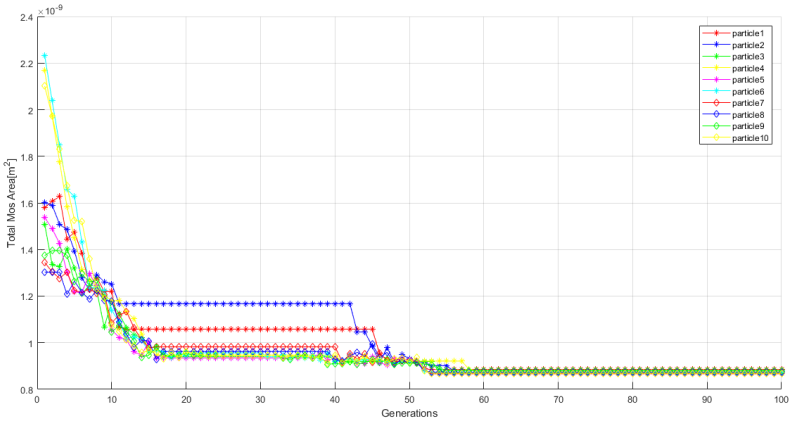
This chapter describes the results of the simulations using PSO, MOL and NSGA-II algorithms for the sizing of topologies presented in chapter 2, that are the Two Stage Miller and the Recycled Folded Cascode OTA, respectively. Usually PSO and MOL algorithms are classified as mono-objective algorithms, despite being mono-objective, these algorithms are not limited to optimizing a single objective, because more targets can be added using constraints. In this work the main objective in addition to sizing the proposed OTAs is to minimize the silicon area and guarantee that transistors operate in strong inversion. In addition a test of process, voltage and temperature variations is performed.

Table 4.1: Characteristics of the two solved problems.

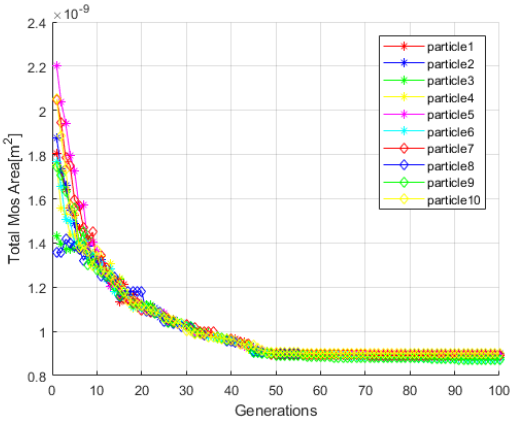
Topology	Miller OTA	Recycled-Folded-Cascode OTA
CMOS technology	0.18μm	0.5μm
Voltage supply [V]	$\pm$0.9	$\pm$1
$C_L[pF]$	5	70
L [μm]	0.36	0.6
Gain [dB]	$\geq$ 60	$\geq$ 60
No. Design variables	4	9
No. constraints	9	17
V_{DSAT}	$>$3	$>$3
No. Individuals	10	10
No. Generations	100	100
No. Runs	10	10

4.1. PSO and MOL Results

In both PSO and MOL the direction of the particle gradually changes to move in direction of the best found positions, looking in its vicinity and potentially discovering better positions according to (3.1), (3.2) and (3.3). In CMOS OTA's sizing, the position of the particle represents the dimensions W that are being updated in order to find the best area. Table 4.1 shows the conditions to execute PSO and MOL to size the MILLER and RFC OTA. The behavior of metaheuristics is not a predictive behavior therefore is required to perform several tests.



(a) PSO



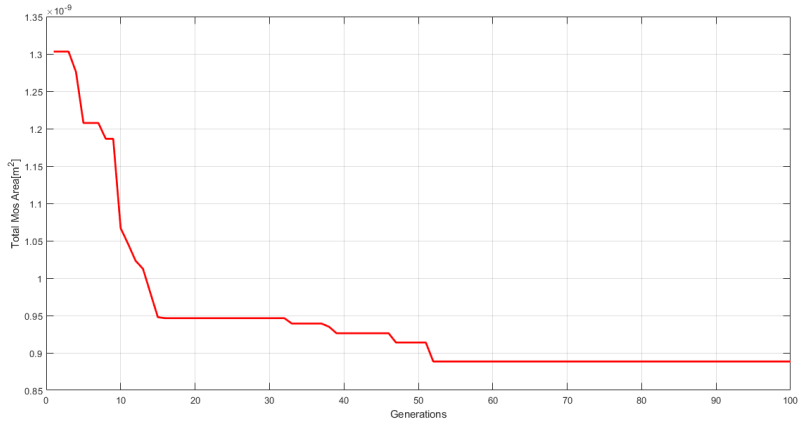
(b) MOL

Figure 4.1: Evolution of the MOS area for each particle applying (a) PSO and (b) MOL.

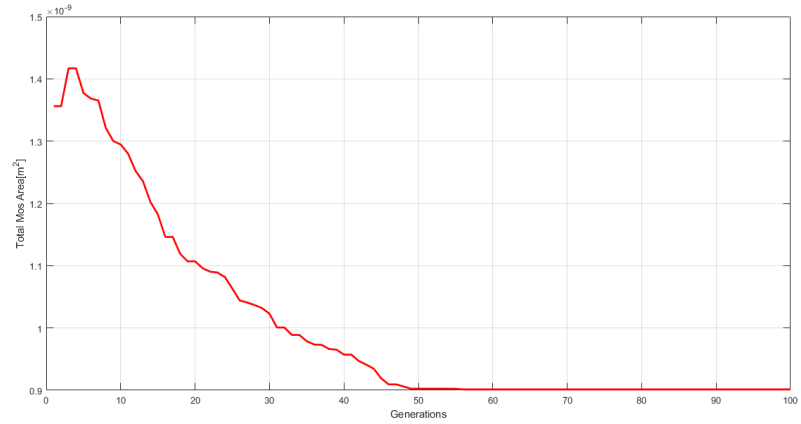
4.1.1. Two Stage Miller OTA Results

The PSO and MOL algorithms are based on a set of particles with a specific position, these positions are updated in each generation in order to find better results in a specific

search space. Figure 4.1 shows the evolution of the area for 10 particles after 100 generations. As is observed the behavior of the PSO's particles tend to follow an individual behavior, this is because the algorithm is designed to find its best position both individually as globally. On the other hand the particles of the MOL algorithm tend to follow the best particle also called global best, this is due to the fact that the MOL algorithm is based on a completely social behavior.



(a) PSO



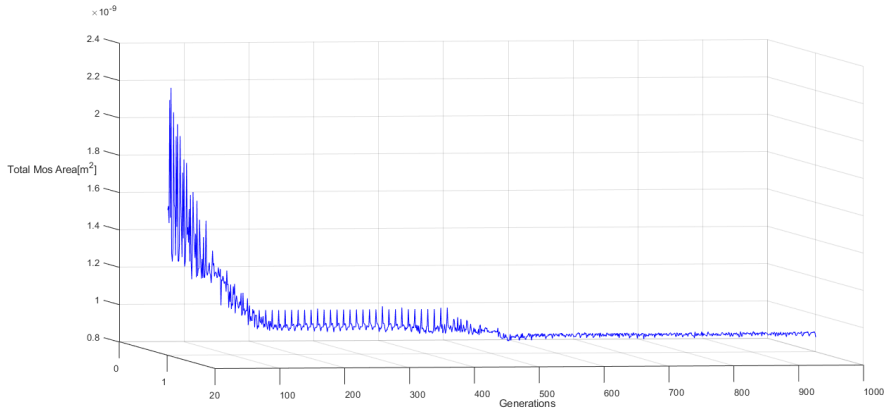
(b) MOL

Figure 4.2: Best global evolution of the particles in sizing the Miller OTA applying (a) PSO and (b) MOL.

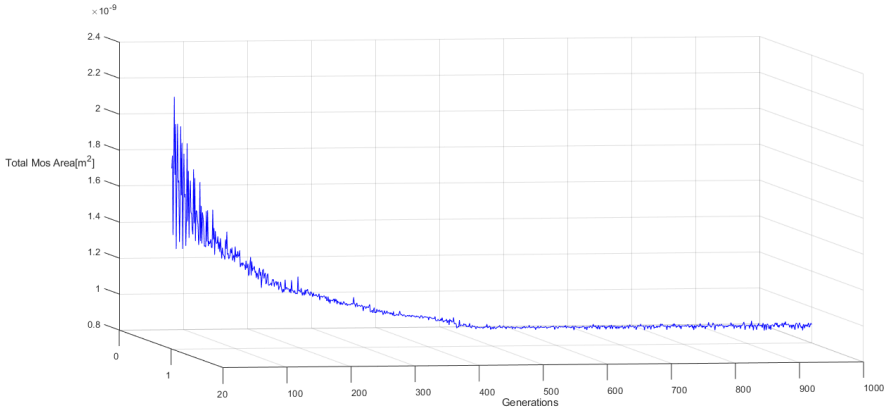
Both PSO and MOL algorithms select the best global particle, this particle is selected with constraints handling. Basically the global best particle is the one that best complies with the constraints, in case two or more particles comply with all of them, the particle with the smallest area is selected. For example; in the first generation in Fig. 4.2(a), the best global is approximately $1300\mu m^2$, which represents the particle8 in Fig. 4.1(a).

This means that in the first generation particle8 was the one that fulfilled the most restrictions, this particle remained until generation 10 where the particle10 obtained better results with an area of $1030\mu m^2$. If none of the particles was better than the best global, this is maintained until another particle has a better performance with the handling constraints. The same analysis is performed for MOL.

Fig. 4.3 allows to appreciate the difference between the feasible solutions provided by PSO and MOL. MOL shows how all the particles tend to follow the global best as the generations increase, while the PSO particles tend to have an unstable behavior until the end of generations. According to the constraint-handling mechanism implemented herein, the best particle is the one that meets the greatest number of constraints, lowest silicon area and differential gain of at least 60dB.



(a) PSO

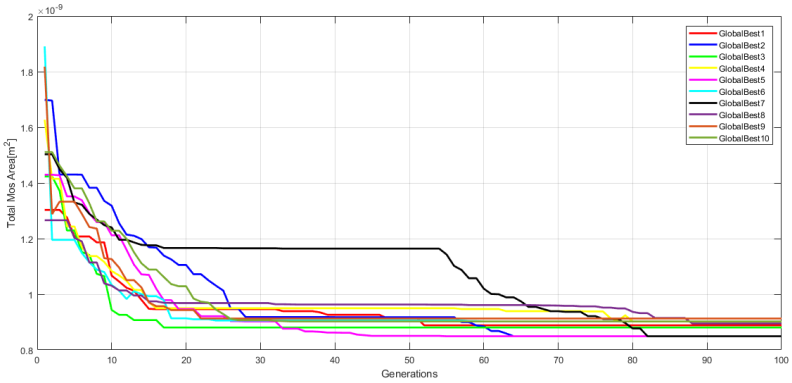


(b) MOL

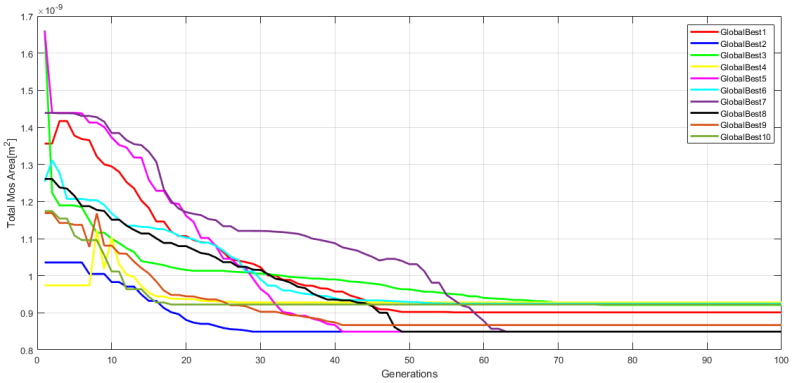
Figure 4.3: Evolution of the particles in sizing the Miller OTA applying (a) PSO and (b) MOL.

Figure 4.4 shows the best evolution of the minimum area solutions for 100 generations during 10 runs. As is observed in these figures, the MOL algorithm finds better results

in a few generations, this is because all the particles follow the best global particle during the whole simulation.



(a) PSO

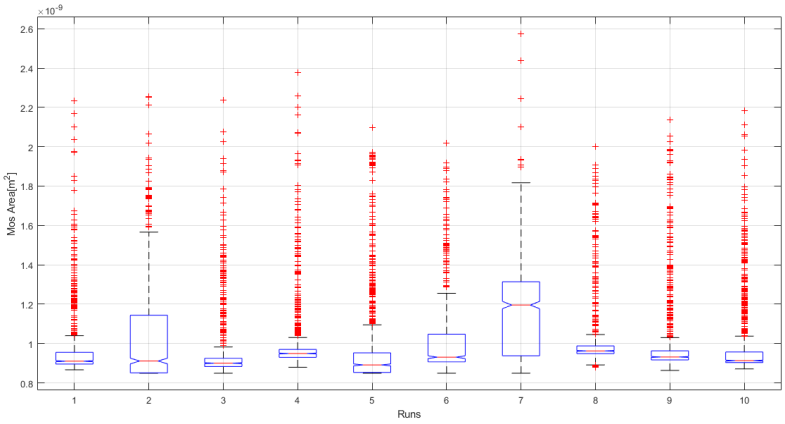


(b) MOL

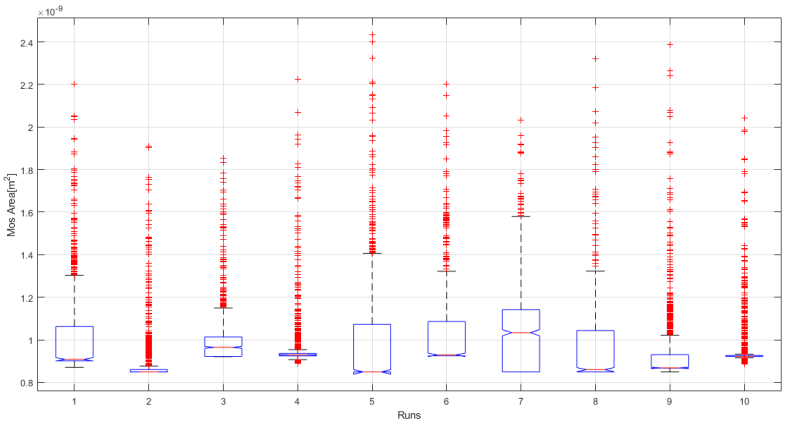
Figure 4.4: Best global evolution of the MOS area for 10 runs.

Metaheuristics are used to obtain a set of feasible solutions, but not always ensure the correct solution, for this reason is important to perform several tests that prove the algorithm is well-tuned. Figure 4.5 shows the averages of the sized solutions. These box graphs describe where the median of the area is located, if the line is not in the center of the box is said that the data is not symmetric. Also shows the outliers, these are represented as red crosses. Figure 4.5 shows how the data provided by the MOL algorithm has a minimum deviation compared with the PSO data. This means that each of the particles of MOL tend to a specific value, while the PSO presents more distributed results within the search space.

These results are detailed in Tables 4.2 and 4.3 for PSO and MOL, respectively. In the last row are shown the values for the objective function that is the total estimated MOS area. The other performances that are listed in the Tables were evaluated during the



(a) PSO



(b) MOL

Figure 4.5: Average of the sized solutions after 10 runs applying (a) PSO and (b) MOL.

optimization process and they are: differential-mode gain A_{OL}, gain-bandwidth product (GBW), common-mode gain A_{CM}, common-mode rejection ratio ($CMRR$), power supply rejection ratio ($PSRR$), positive slew rate ($SR+$), negative slew rate ($SR-$), maximum input voltage (Vin_{max}), minimum input voltage (Vin_{min}), width of the MOS transistors (M1 and M2 encoded by W1), width of the P-type MOS transistors W2, width of the N-type MOS transistors W3, length of all MOS transistors L, and current bias I_b.

To verify the results presented in Tables 4.2 and 4.3 are correct, one of the runs is selected to carry out the characterization individually for each of measurements. Figures 4.6 and 4.7 show the results for the differential mode gain, GBW, and the common mode gain, respectively.

Table 4.2: Electrical characteristics and feasible W/L sizes of the Miller OTA guaranteeing DCOP conditions of the MOS transistors applying PSO for 10 runs. The best values are highlighted in bold face.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Aol[dB]	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**	**60**
GB[MHz]	**26**	21	23	26	21	**26**	21	**26**	**26**	**26**
Acm[dB]	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**
CMRR[dB]	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**
PSRR+[dB]	**70**	**70**	10	69	**70**	69	**70**	**70**	69	69
PSRR-[dB]	**129**	118	108	120	118	**129**	118	110	120	120
SR+[v/us]	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
SR-[v/us]	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
W1[M1,M2][um]	34	27	30	34	27	27	27	36	35	34
W2[M4,M5][um]	27	27	28	28	27	32	27	27	27	28
W3[M6][um]	54	54	56	56	54	64	54	54	54	56
W4[M3,M7][um]	54	54	60	54	54	54	54	54	58	54
W5[Mbias][um]	27	27	30	27	27	27	27	27	29	27
L[um]	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
Ib[uA]	50	40	44	50	40	50	40	50	50	50
Total MOS area[um^2]	888	**849**	880	902	**849**	902	**849**	895	912	902

Table 4.3: Electrical characteristics and feasible W/L sizes of the Miller OTA guaranteeing DCOP conditions of the MOS transistors applying MOL for 10 runs. The best values are highlighted in bold face.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Aol[dB]	60	60	**61**	60	60	60	60	60	60	60
GB[MHz]	**26**	21	25	**26**	21	22	21	21	23	**26**
Acm[dB]	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**	**-3**
CMRR[dB]	63	63	**64**	63	63	63	63	63	63	63
PSRR+[dB]	69	**70**	**70**	**70**	**70**	**70**	**70**	**70**	**70**	**70**
PSRR-[dB]	**120**	118	115	105	118	105	118	118	103	110
SR+[v/us]	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
SR-[v/us]	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
W1[M1,M2][um]	27	27	27	32	27	27	27	27	31	34
W2[M4,M5][um]	32	27	33	30	27	33	27	27	27	30
W3[M6][um]	64	54	66	60	54	66	54	54	54	60
W4[M3,M7][um]	54	54	54	56	54	54	54	54	54	54
W5[Mbias][um]	27	27	27	28	27	27	27	27	27	27
L[um]	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
Ib[uA]	50	40	50	50	40	42	40	40	43	50
Total MOS area[um^2]	901	**849**	921	927	**849**	921	**849**	**849**	866	922

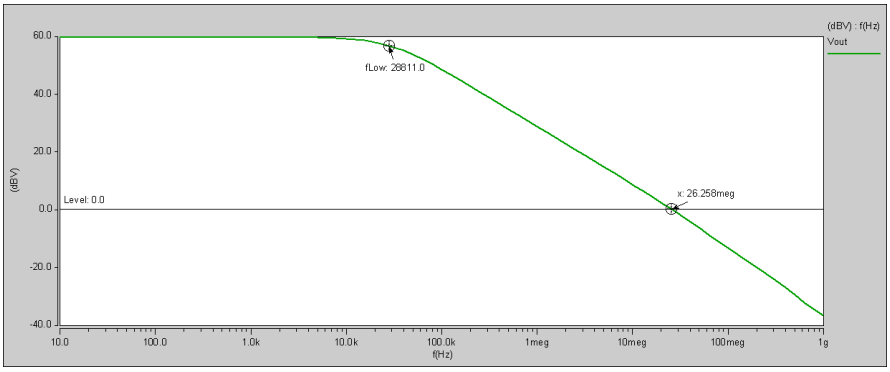


Figure 4.6: Algorithm results to the A_{OL} and GB.

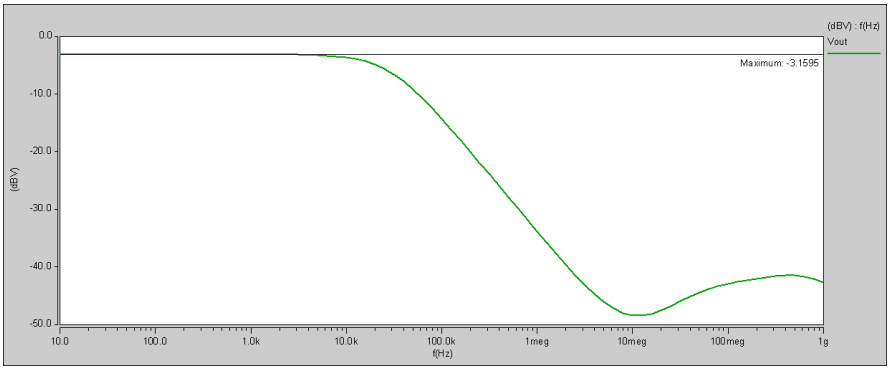


Figure 4.7: Algorithm results to the A_{CM}.

CMRR is one of the most important characteristics of an OTA since is a measure of the rejection offered by the configuration to the common voltage input. Figure 4.8 shows the CMRR measurement.

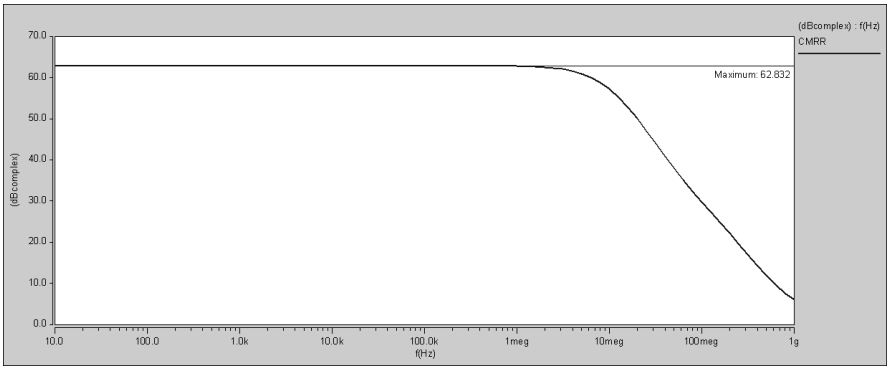


Figure 4.8: Algorithm results to the $CMRR$.

Figures 4.9 and 4.10 show the positive and negative PSRR.

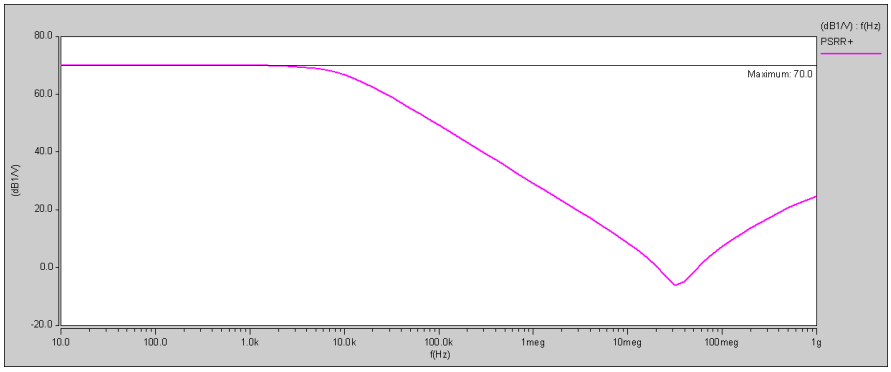


Figure 4.9: Algorithm results to the $PSRR+$.

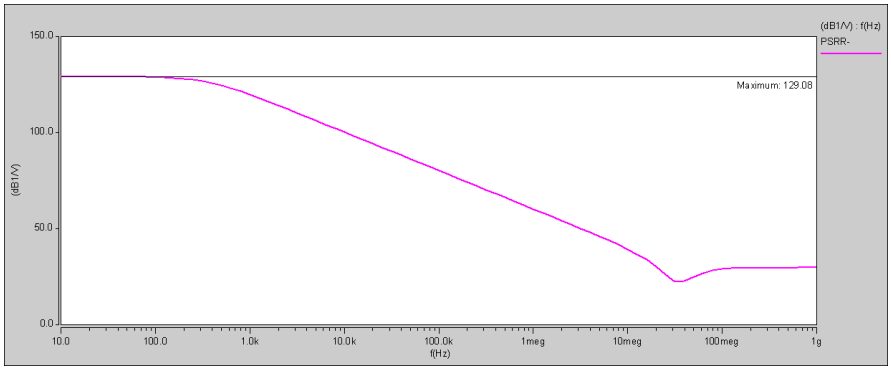


Figure 4.10: Algorithm results to the $PSRR-$.

Finally, Figure 4.11 shows the results of the time analysis for the positive and negative Slew Rate.

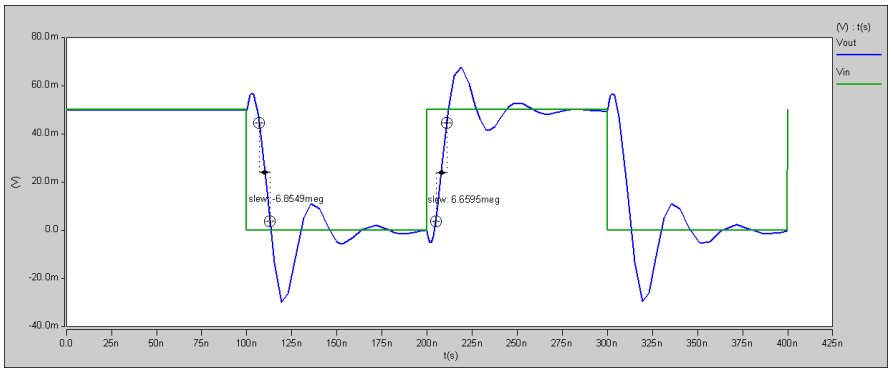
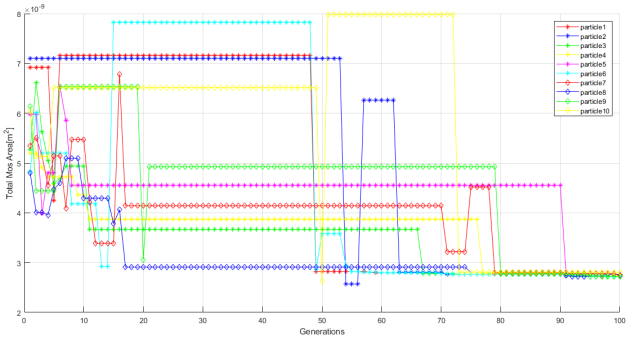


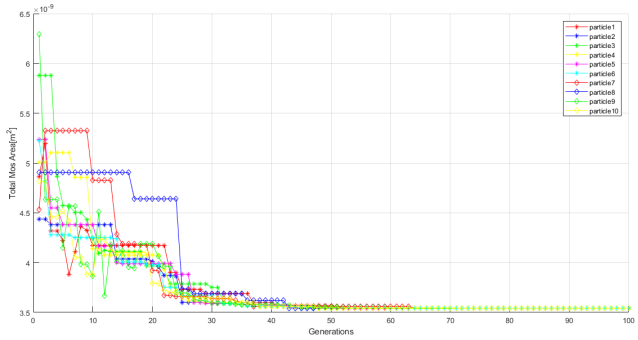
Figure 4.11: Algorithm results to the $SR\pm$.

4.1.2. Recycled Folded Cascode OTA Results

The RFC-OTA shown in Fig. 2.16 is sized by applying PSO and MOL algorithms to minimize the silicon area and ensure that the MOS transistors operate in the desired DCOP region. Figure 4.12 shows the evolution of the particles in each generation. It should be noted that the encoding of this circuit must have more design variables, W and L compared to the two-stages OTA Miller, because it has more MOS transistors. In this manner, eight widths W are encoded as shown in Tables 4.4 and 4.5. In the simulations for the RFC OTA design, it was considered a factor $k = 3$.



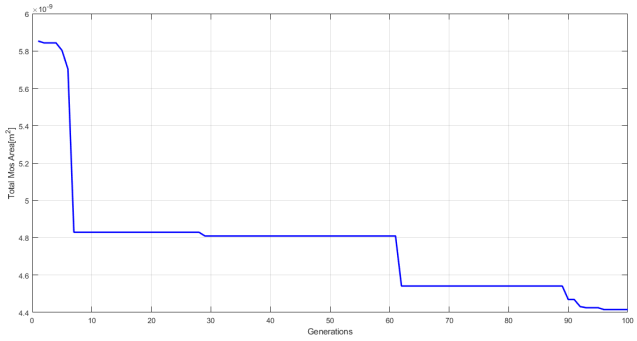
(a) PSO



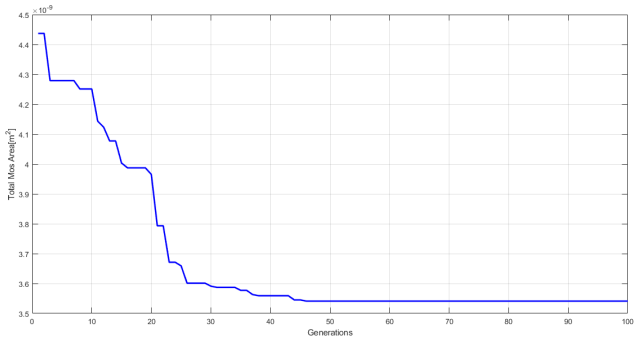
(b) MOL

Figure 4.12: Evolution of the MOS area for each particle applying (a) PSO and (b) MOL.

Figure 4.13 shows the evolution of the best global feasible solution. The behavior of the area evolution of the RFC-OTA is very similar to the two-stages OTA Miller. In this case, it can be appreciated that all the particles tend to the best global after 20 generations for the MOL algorithm, and the PSO tends to have unpredictable behavior. The particle's behavior is mainly due to the constraints handling. Usually no particle finds a feasible solution from the beginning of the simulation, however with the update equations the algorithm approaches the best solutions with the course of generations.

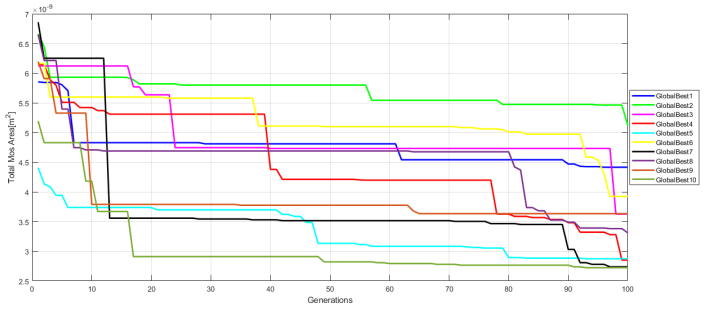


(a) PSO

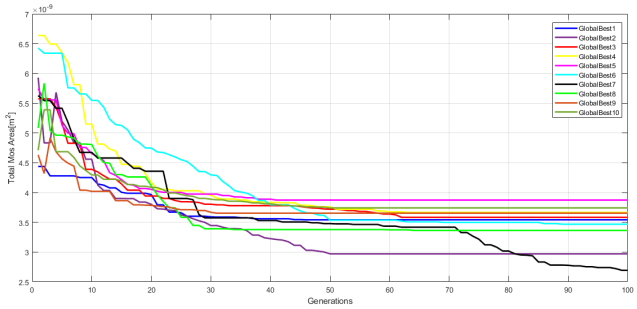


(b) MOL

Figure 4.13: Best global evolution of the particles in sizing the RFC OTA applying (a) PSO and (b) MOL.



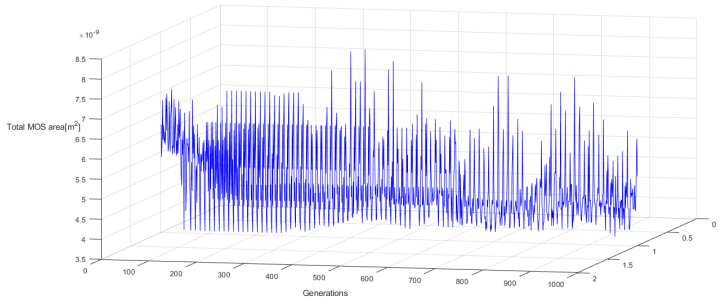
(a) PSO



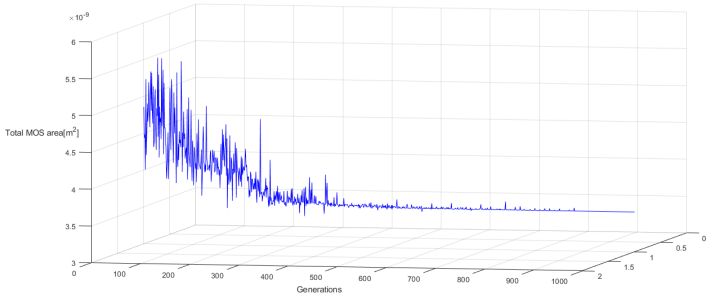
(b) MOL

Figure 4.14: Best global evolution of the MOS area for 10 runs (a) PSO and (b) MOL.

PSO and MOL can be applied in large spaces of candidate solutions since are assumed few hypotheses about the problem to be optimized. Nevertheless, like all metaheuristics, PSO and MOL does not guarantee obtaining an optimal solution in all cases, as shown in Figure 4.14. Due to the circuit complexity the algorithms require more time to find feasible solutions and satisfy each of the specified constraints. According to figure 4.15(a), is observed how the particles change from one solution to another very different without marking a certain tendency in the optimization of the area. On the other hand MOL algorithm keeps the same behavior despite increasing the circuit complexity, always follows the best particle from the beginning.



(a) PSO



(b) MOL

Figure 4.15: Evolution of the particles in sizing the RFC OTA applying (a) PSO and (b) MOL.

Another important difference with respect to the results of the algorithm for the Miller OTA is the variation of the average area amount runs. Figure 4.5 shows how the average of all the runs stays in an approximate area, while the average area for the RFC OTA changes from one run to another as observed in Fig. 4.16. The results of the RFC OTA measurements are shown in Tables 4.4 and 4.5. To verify that the data resulting from the algorithm is correct, measurements of the characteristics of the RFC OTA are carried out. The values of W and I_b are the data in column 2 of table 4.4.

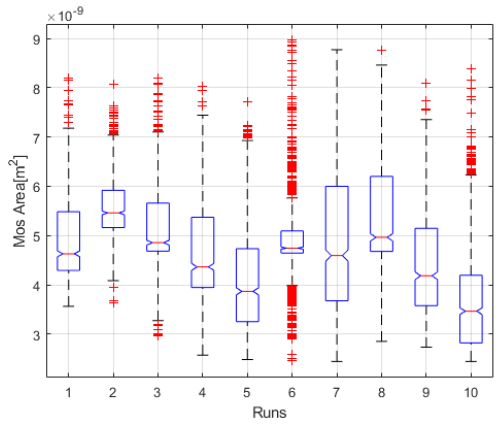
The differential mode gain and GBW of the RFC OTA are depicted in Figure 4.17. Figures 4.18 and 4.19 show the common mode gain and the CMRR, respectively.

Table 4.4: Electrical characteristics and feasible W/L sizes of the RFC OTA guaranteeing DCOP conditions of the MOS transistors applying PSO for 10 runs. The best values are highlighted in bold face.

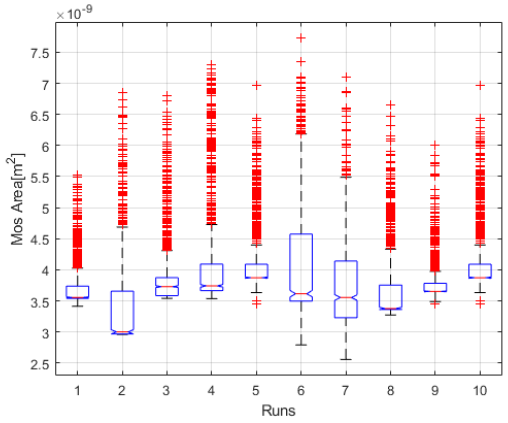
	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Aol[dB]	73	**77**	71	74	75	73	75	72	71	75
GB[KHz]	446	**758**	347	246	412	449	382	357	422	382
Acm[dB]	-32	**-36**	-32	-26	-32	-32	-29	-31	-33	-29
CMRR[dB]	105	**113**	103	100	107	105	104	103	104	104
PSRR+[dB]	93	69	98	**102**	80	100	98	98	98	110
PSRR-[dB]	**100**	**100**	97	97	97	95	96	91	90	96
SR+[v/us]	**0.15**	0.14	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**
SR-[v/us]	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**
W1[M1a,M2a][um]	18	60	12	12	12	25	25	12	15	25
W2[M1b,M2b][um]	12	19	13	17	13	14	13	13	13	13
W3[M3a,M4a][um]	12	60	30	13	16	13	13	13	12	13
W4[M3b,M4b][um]	12	48	46	31	34	12	14	17	13	14
W5[M3c,M4c][um]	12	12	13	13	12	12	12	12	12	12
W6[M5,M6][um]	36	12	12	12	13	12	14	13	13	14
W7[M7,M8][um]	60	12	12	13	13	12	13	16	12	13
W8[M9,M10][um]	27	12	26	12	12	60	12	41	60	12
L[um]	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
Ib[uA]	10	20	10	10	20	10	10	10	10	10
Total Mos Area[um^2]	4416	5118	3630	2852	2874	3924	**2738**	3310	3634	**2738**

Table 4.5: Electrical characteristics and feasible W/L sizes of the RFC OTA guaranteeing DCOP conditions of the MOS transistors applying MOL for 10 runs. The best values are highlighted in bold face.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
Aol[dB]	**76**	71	73	71	**76**	75	67	75	71	73
GB[KHz]	476	325	**612**	411	457	364	561	320	411	457
Acm[dB]	-31	-31	**-34**	-33	-31	-30	-20	-28	-33	-31
CMRR[dB]	107	102	**107**	104	**107**	105	87	103	104	104
PSRR+[dB]	75	79	74	79	70	**81**	78	79	79	70
PSRR-[dB]	97	97	99	101	**104**	100	100	96	101	**104**
SR+[v/us]	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**	**0.14**
SR-[v/us]	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**	**0.15**
W1[M1a,M2a][um]	28	12	32	16	37	13	20	15	16	37
W2[M1b,M2b][um]	12	15	12	13	32	12	12	12	13	32
W3[M3a,M4a][um]	32	12	17	32	17	27	14	21	32	17
W4[M3b,M4b][um]	29	19	12	37	30	53	12	32	37	30
W5[M3c,M4c][um]	12	12	13	13	12	12	13	13	13	13
W6[M5,M6][um]	17	12	13	12	12	12	13	12	12	14
W7[M7,M8][um]	13	12	12	12	12	16	12	13	12	13
W8[M9,M10][um]	13	26	37	28	12	12	12	12	28	14
L[um]	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
Ib[uA]	12	10	13	10	15	14	12	10	10	15
Total Mos Area[um^2]	3542	2940	3584	3666	3874	3468	**2680**	3364	3666	3734



(a) PSO



(b) MOL

Figure 4.16: Average of the sized solutions after 10 runs applying (a) PSO and (b) MOL.

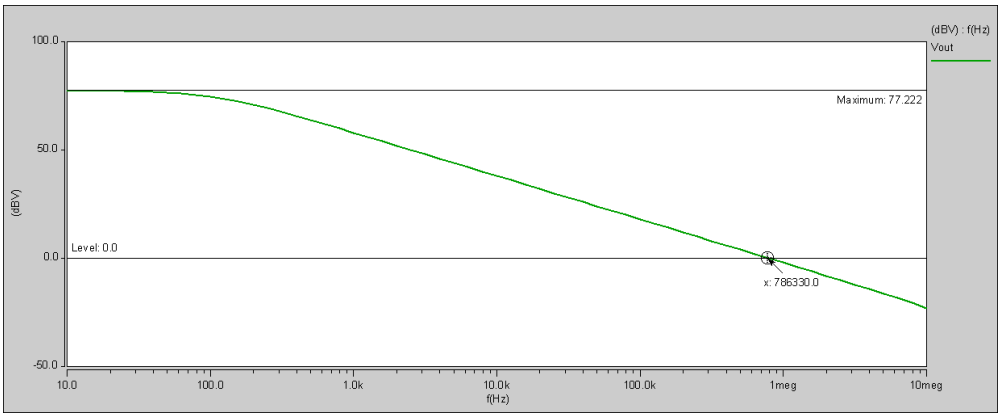


Figure 4.17: Algorithm results to the A_{OL} and GB.

Subsequently, the positive and negative PSRR Figures are presented in 4.20 and 4.21,

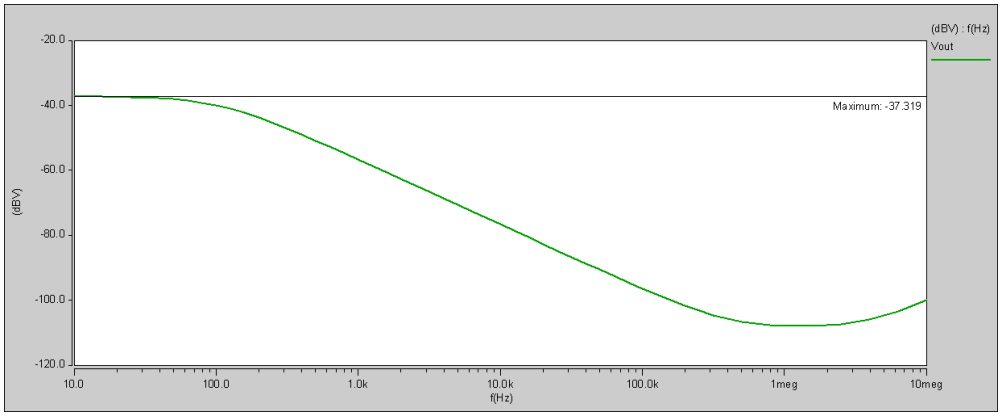


Figure 4.18: Algorithm results to the A_{CM}.

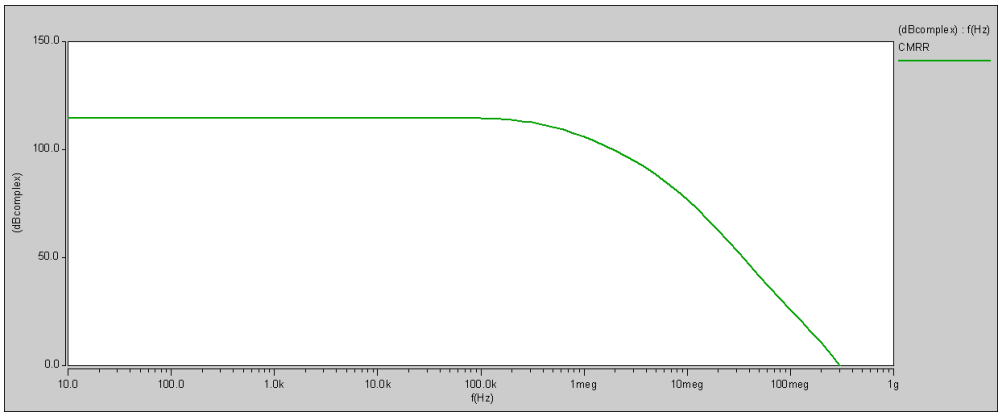


Figure 4.19: Algorithm results to the $CMRR$.

respectively. Finally, in Figure 4.22, the positive and negative SR measurements are shown.

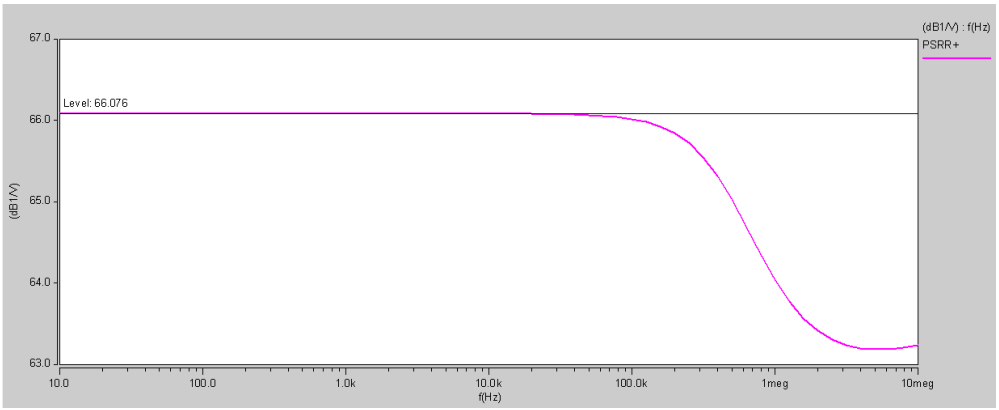


Figure 4.20: Algorithm results to the $PSRR+$.

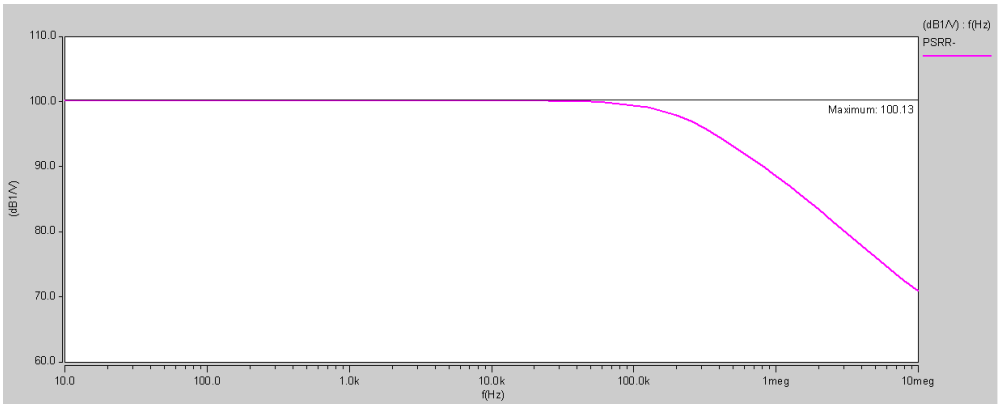


Figure 4.21: Algorithm results to the $PSRR-$.

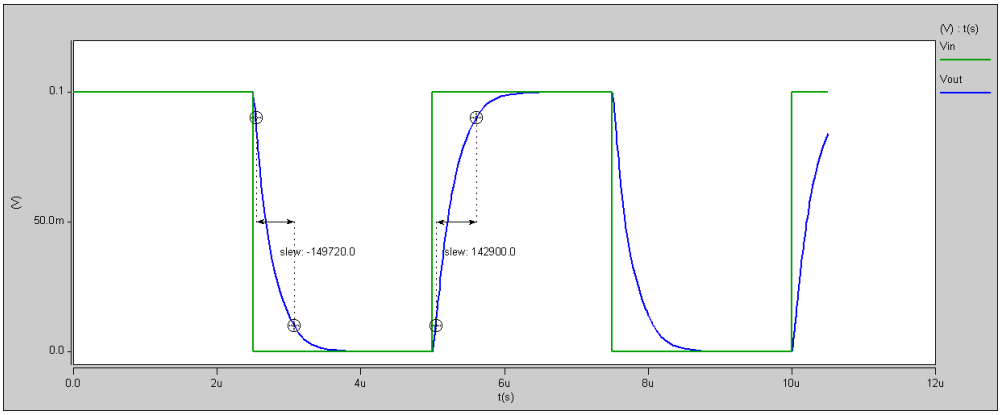


Figure 4.22: Algorithm results to the $SR\pm$.

4.2. NSGA-II Results

NSGA-II is one of the best multi-objective algorithms to solve optimization problems. This test is performed for the same topologies presented in section 2, in order to compare the behavior of mono-objective algorithms such as the PSO and MOL with a multi-objective algorithm such as the NSGA-II. One of the advantages of using multi-objective algorithms is the maximization or minimization of two objectives at the same time without use constraints to truncate the results.

NSGA-II extracts several Pareto fronts composed of individuals to which an adaptation value is assigned, also called ranking. In the first Pareto front there are the non-dominated individuals, these individuals are assigned a high adaptation value. Meanwhile, the dominated individuals are ranked in the same way to identify the Pareto sub-front. This process continues until all individuals are classified in fronts.

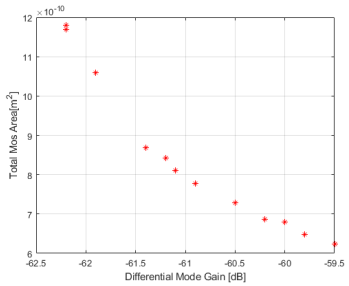
4.2.1. Two Stage Miller OTA Results

Figure 4.23 shows the Pareto fronts for 10 runs, these Pareto fronts show all feasible solutions over 100 generations with a population of 20 individuals. Is important to specify that in order to propose two objectives, there must be a conflict between them to improve the performance of the algorithm. In general, an evolutionary algorithm is encoded to minimize a function, therefore to be able to increase the differential mode gain as an objective is necessary to invert the function.

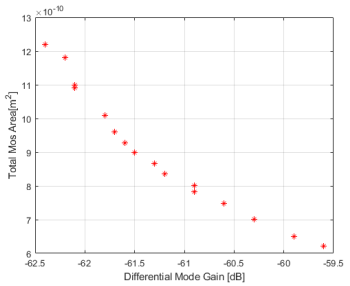
Table 4.6: Electrical characteristics and feasible W/L sizes of the Miller OTA guaranteeing DCOP conditions of the MOS transistors applying NSGA-II. The best values are highlighted in bold face.

Aol[dB]	60	60.2	60.5	61.4	**62.4**
GB[MHz]	**25**	**25**	24	24	24
Acm[dB]	**-3**	**-3**	**-3**	**-3**	**-3**
CMRR[dB]	63	63.2	63.5	64.4	**65.4**
PSRR+[dB]	**70**	**70**	**70**	**70**	**70**
PSRR-[dB]	**105**	98	98	98	98
SR+[v/us]	**6**	**6**	**6**	**6**	**6**
SR-[v/us]	**6**	**6**	**6**	**6**	**6**
W1[M1,M2][um]	27	29	29	30	27
W2[M4,M5,M6][um]	27	33	38	55	87
W3[M3,M7,Mbias][um]	27	27	27	27	27
L[um]	0.36	0.36	0.36	0.36	0.36
Ib[uA]	50	50	50	50	50
Total Mos Area[um^2]	**622**	689	729	899	1220

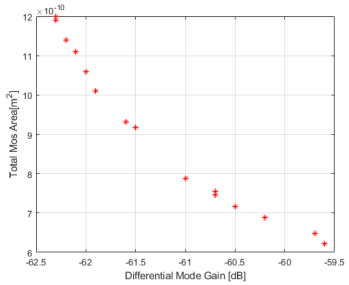
A chromosome is formed by 4 genes $w1$, $w2$, $w3$ and I_b, where the objective functions are the minimization of the area and the maximization of the differential mode gain, in addition the region of operation of the transistors is guaranteed. Fig. 4.24 shows the Pareto front formed by the 5 best solutions of all the Paretos front shown in Figure 4.23. Table 4.6 summarizes the electrical characteristics and feasible W/L sizes of the 5 best solutions.



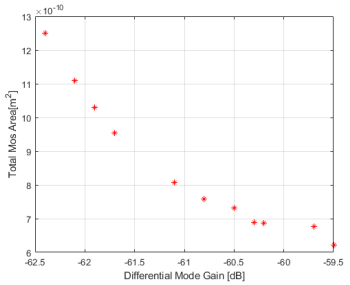
(a) RUN1



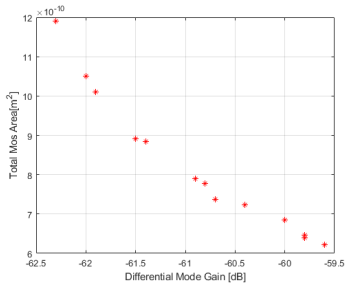
(b) RUN2



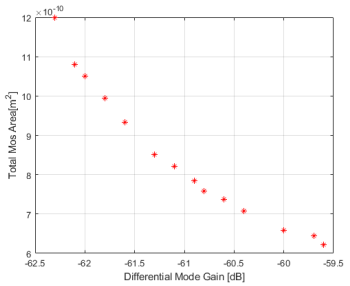
(c) RUN3



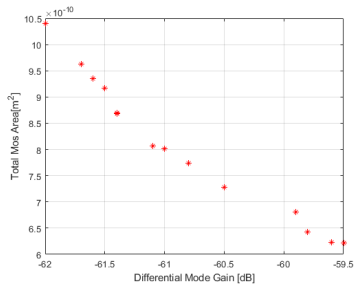
(d) RUN4



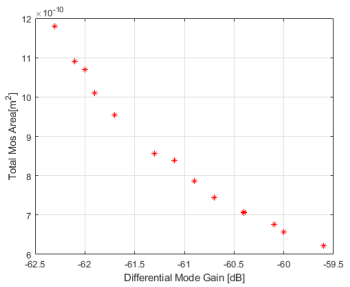
(e) RUN5



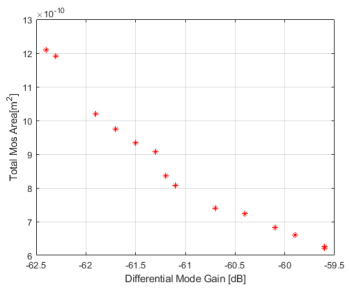
(f) RUN6



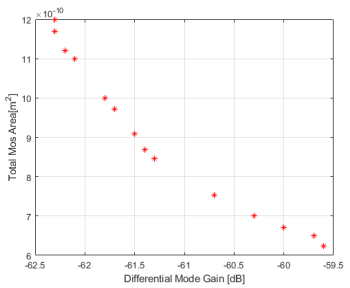
(g) RUN7



(h) RUN8



(i) RUN9



(j) RUN10

Figure 4.23: Pareto Fronts of the Two Stage Miller OTA for 10 runs.

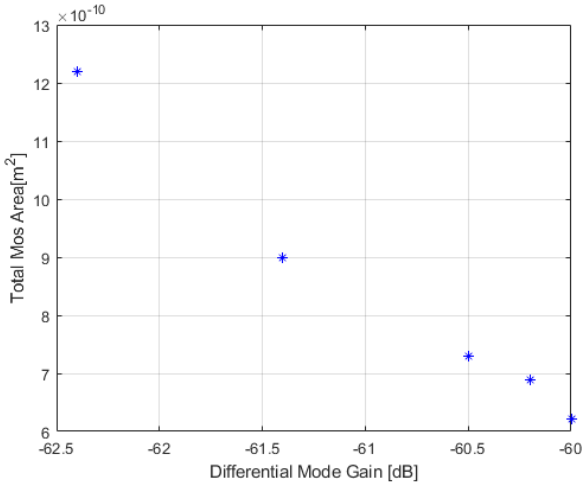


Figure 4.24: Pareto Front of 5 best solutions.

4.2.2. Recycled Folded Cascode OTA Results

To perform the RFC OTA sizing is necessary to create a chromosome with 9 genes $w1$, $w2$, $w3$, $w4$, $w5$, $w6$, $w7$, $w8$ and I_b, considering a factor of $k = 3$ for the current mirrors. NSGA-II algorithm tends to find better solutions with greater number of generations, this also depends on the value of the genetic operators. The 5 best solutions of the 10 Pareto fronts shown in Figure 4.26 are plotted in Figure 4.25. Results are shown in Table 4.7.

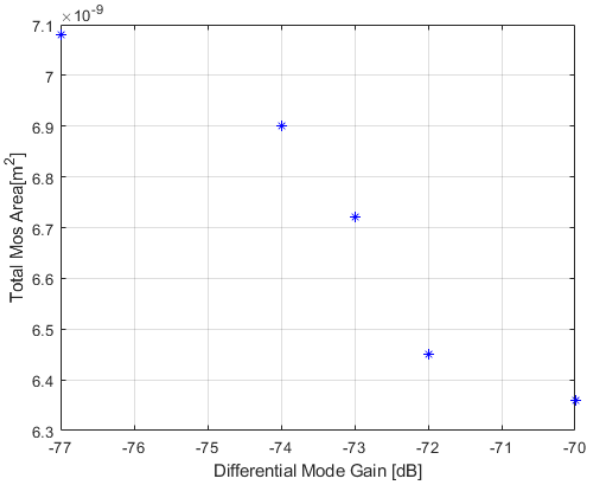


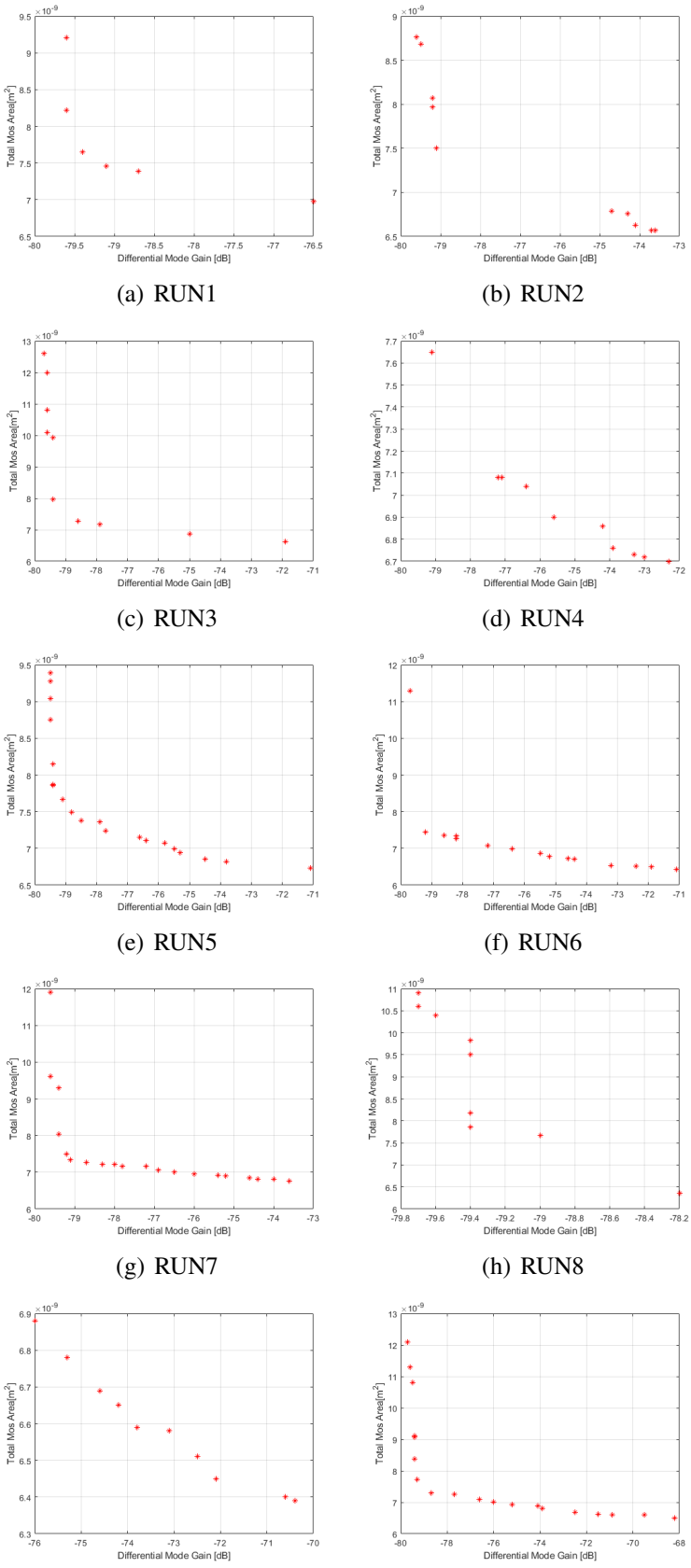
Figure 4.25: Pareto Front of 5 best solutions.

Figure 4.26 shows the Pareto fronts for 10 runs, as shown in this Figure the number of feasible solutions does not match with the number of individuals,since not all indivi-

duals find the best solution in that number of generations. In addition Pareto fronts are not always ideally formed, this is usually due to the crossover and mutation probability values.

Table 4.7: Electrical characteristics and feasible W/L sizes of the RFC OTA guaranteeing DCOP conditions of the MOS transistors applying NSGA-II. The best values are highlighted in bold face.

Aol[dB]	70	72	73	74	**77**
GB[MHz]	1.02	1.19	0.9	1	**1.3**
Acm[dB]	-40	-42	-38	-34	**-48**
CMRR[dB]	110	114	111	107	**125**
PSRR+[dB]	53	55	40	39	**75**
PSRR-[dB]	**94**	**94**	80	80	**94**
SR+[v/us]	0.22	0.25	**0.4**	**0.4**	0.24
SR-[v/us]	0.22	0.25	**0.4**	**0.4**	0.24
W1[M1a,M2a][um]	61	62	66	76	68
W2[M1b,M2b][um]	60	60	60	60	60
W3[M3a,M4a][um]	60	60	60	60	61
W4[M3b,M4b][um]	63	73	89	89	119
W5[M3c,M4c][um]	60	60	60	60	61
W6[M5,M6][um]	**60**	60	60	60	62
W7[M7,M8][um]	60	60	61	61	61
W8[M9,M10][um]	61	61	60	60	60
L[um]	0.6	0.6	0.6	0.6	0.6
Ib[uA]	20	27	45	45	45
Total Mos Area[um^2]	**6360**	6450	6720	6900	7080



(a) RUN1 (b) RUN2

(c) RUN3 (d) RUN4

(e) RUN5 (f) RUN6

(g) RUN7 (h) RUN8

(i) RUN9 (j) RUN10

Figure 4.26: Pareto Fronts of the RFC OTA for 10 runs.

Chapter 5
Design performance and results

5.1. Layout

In this section the layout of a Two Stage Miller OTA is performed. The first step to achieve a good layout is to perform a floorplaning in which all the transistors, dimensions, interdigitated, sensitivity, multiplicities and position are considered. This step is the most important and difficult because it requires time, creativity and experience from the designer to avoid layout mismatches. This step usually requires more time than the others, for this reason the use of integer values that are multiples of the lambda is facilitates the calculation of multiplicities, interdigitated and also reduces the design time.

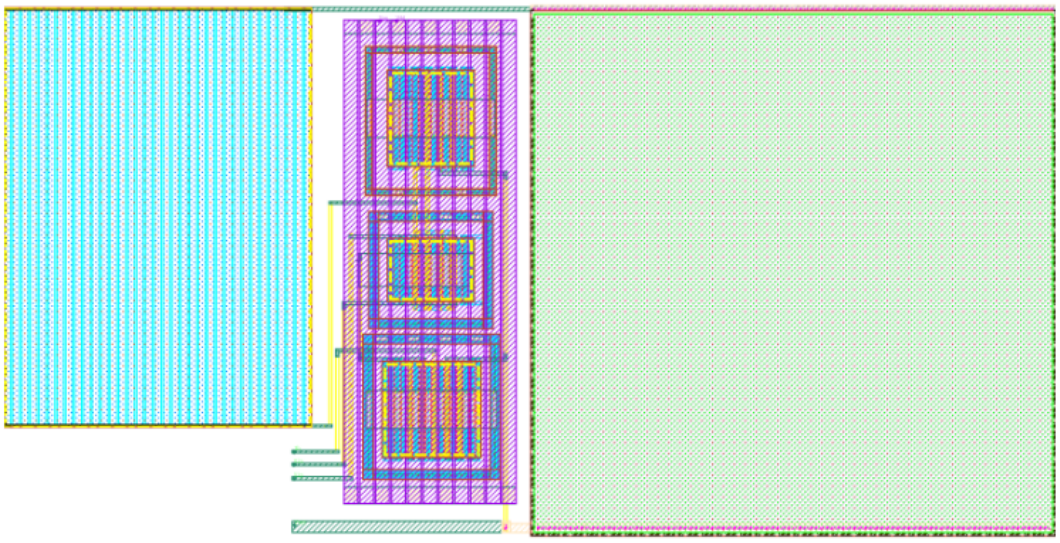


Figure 5.1: Layout of the Miller OTA

Another important aspect is the symmetry, this allows to reduce the variations that can affect the circuit in the fabrication process, reduces the mismatch between the transistors and ensures the voltage or current is distributed equally to the corresponding transistors. Fig. 5.1 shows the implemented layout. One of the main objectives is the minimization of the area, therefore transistors with small areas allow to guarantee a smaller silicon area, despite not considering multiplicities, guard rings and rules of technology. For the layout the electromigration rules were considered, which was not a problem since the polarization current is small enough for the metal's widths. Figure 5.1 also shows the use of a power grid, which allows to distribute all the power to the entire circuit.

Subsequently is important to perform the post-layout simulations, this serves to verify that the operation of the IC remains the same as the main design. The post-layout measurements of each of the characteristics of the Miller OTA are shown in Figures [5.2-5.7].

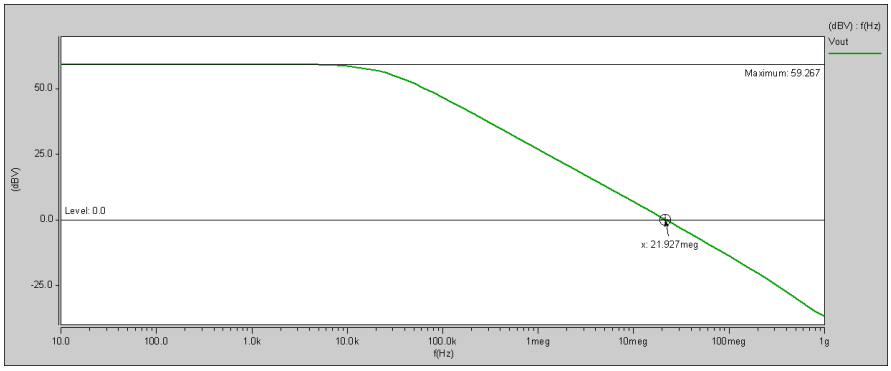


Figure 5.2: Post-Layout results to the A_{OL}.

The figure 5.2 shows how the differential mode gain varies a bit compared to the results of the main circuit. The GBW was the feature that was most affected, due to parasitic capacitances. The other measurements are very close to the previously obtained measurements.

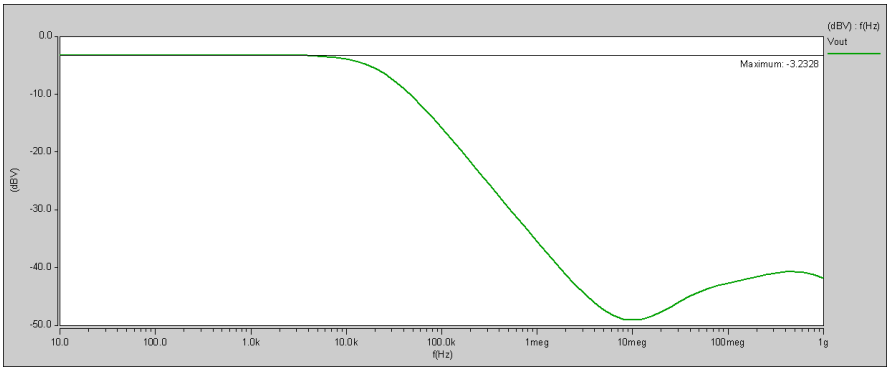


Figure 5.3: Post-Layout results to the A_{CM}.

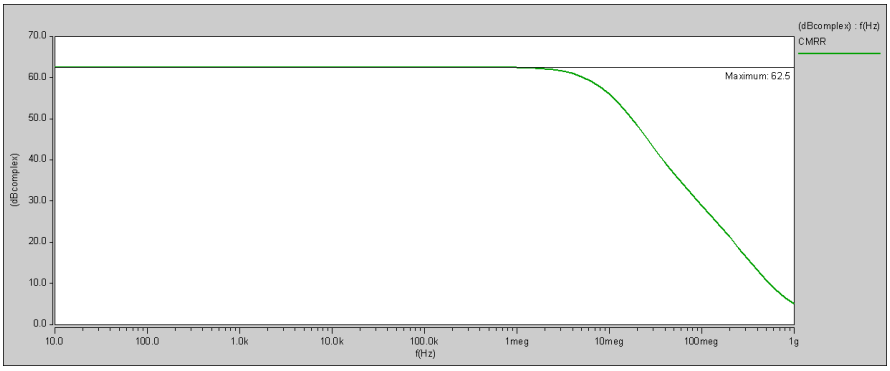


Figure 5.4: Post-Layout results to the $CMRR$.

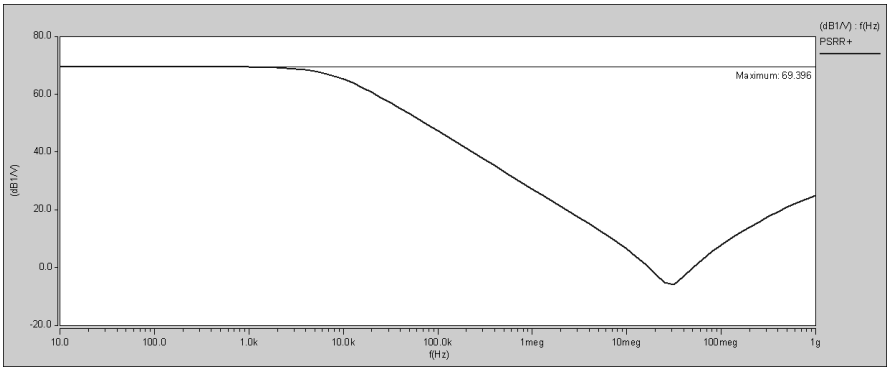


Figure 5.5: Post-Layout results to the $PSRR+$.

5.2. PVT Variations

Nowadays, the industry that studies the design of analog integrated circuits faces a lot of challenges, due to the high demand of electronic devices. The design of analog

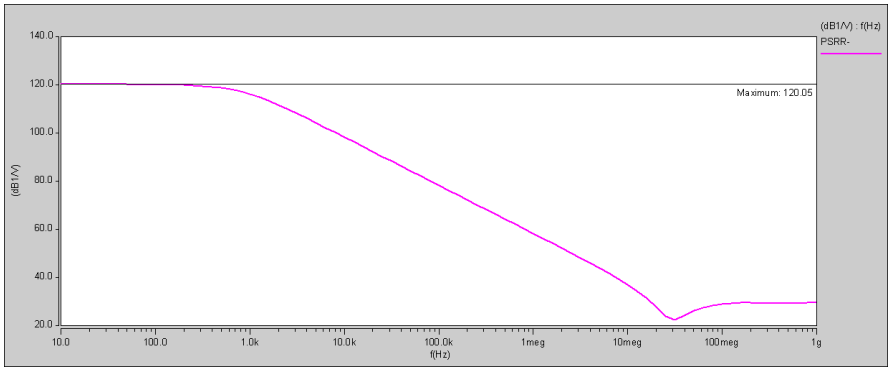


Figure 5.6: Post-Layout results to the $PSRR-$.

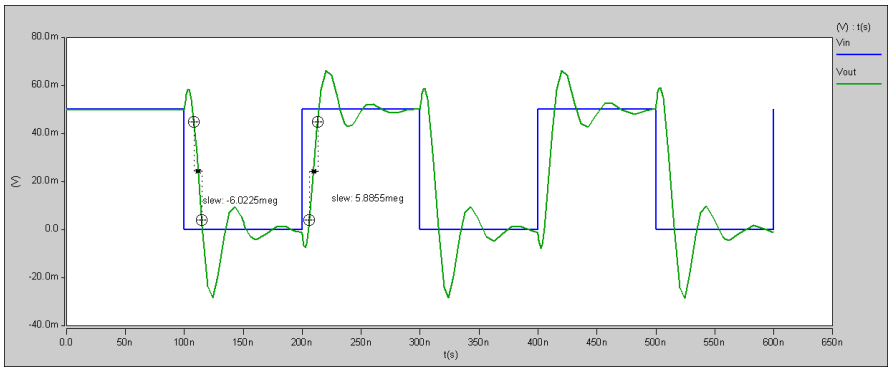


Figure 5.7: Post-Layout results to the $SR+$ and $SR-$.

integrated circuits is usually carried out under ideal operating conditions, therefore in the laboratory is not a problem. But when the circuit is subjected to different conditions they are vulnerable to variations. For this reason is important to perform PVT tests on integrated circuits, this tests are used to determine how robust a circuit is under different variations tests. Usually, these variations can be classified into three main groups: process, voltage and temperature.

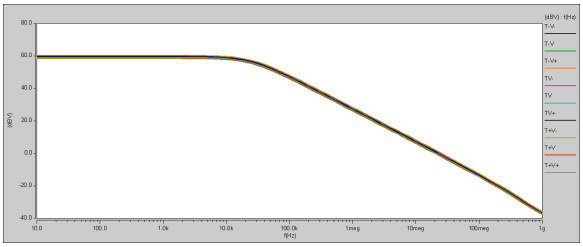
- Process variations depend on the corners within which a device may vary. Generally the integrated circuits are manufactured in silicon wafers, therefore these variations refer to the differences in the doping of the devices. The corner's variation makes the devices work very fast or very slow depending on the fluctuations in the amount of doping, this is due to the dispersion of the current carrying particles due to thermal heating. There are 5 corners: the typical value (TT), fast-fast (FF) which refers to the best case and the worst case slow-slow (SS). The other two corners are the combination of the previous two corners slow-fast (SF) and fast-slow (FS).

- The voltage variations are mainly due to the fact that the supply voltages do not provide the specified nominal voltage. Therefore is important to simulate

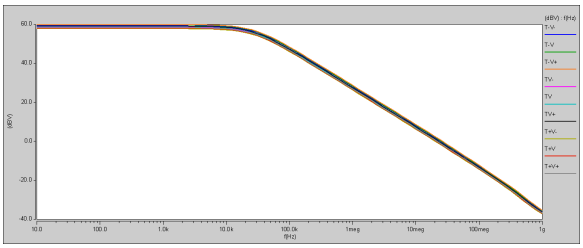
a variation in the supply voltage of $\pm 10\,\%$ of the nominal value of the power supply.

- Integrated circuits often operate in environments where the temperature is not stable, also is important to consider that these circuits are surrounded by others that dissipate energy in form of heat. Usually the design of integrated circuits is done at a temperature of 60^{o} and not a 25^{o} as expected. The temperature corners that are commonly used are 60 ^{o}, -20^{o} and 100^{o}.

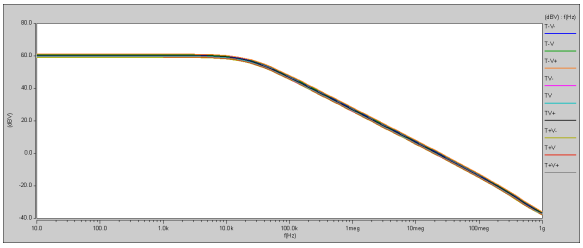
Figure 5.8 shows the PVT analysis for each of the corners TT, FF, SS, FS and SF. For each corner a temperature of -20^{o}, 60^{o} and 120^{o} is used with a variation in the supply voltage of $\pm 10\,\%$.



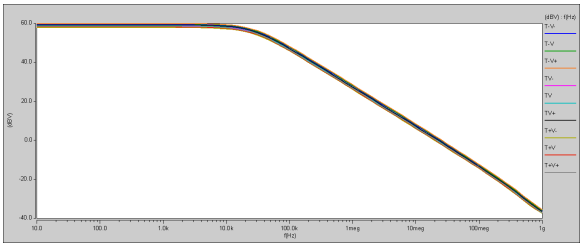
(a) TT



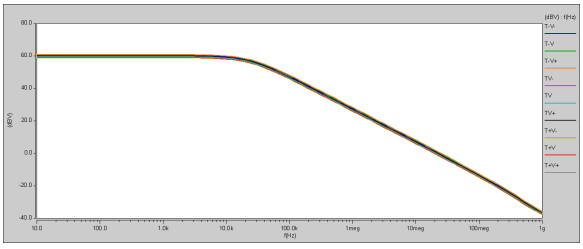
(b) FF



(c) SS



(d) FS



(e) SF

Figure 5.8: PVT analysis for each corner.

Chapter 6
Conclusions and Future work

6.1. Conclusions

Analog integrated circuit design is a process that can be optimized and automated, since is one of the most complex processes in the electronics area, this is due to the large number of constraints and specifications that must be satisfied. This work mainly focuses on transconductance operational amplifiers due to its importance circuits in the development of converters and amplification stages. The design of OTAs becomes a complicated task when is necessary to improve two or more characteristics, since there are many trade-offs which compromise the operation of the circuit. In this work, a hand-drawn design of an OTA Miller is developed using the quadratic model's equations to compare the results with those obtained from the algorithms.

In addition, it was demonstrated that both mono and multi-objective metaheuristics based on population and evolutionary algorithms such as PSO, MOL and NSGA-II serve to optimize CMOS OTAs based on the design variables. In this work, restrictions were used to guide the algorithm to a fast convergence and to modify the mono-objective algorithms such as the PSO and MOL to be used with more objectives.

To obtain more accurate results about the circuit's perfomance, the code of the algorithms was modified to achieve the corresponding simulations with SPICE. This represents a great advantage over the commonly performed design method, since the results are provided directly from the SPICE .lis output files. The design variables are scaled by the technology's lambda within SPICE, therefore the assignment of integers values to the design variables in the algorithm for transistor's sizing is a matter of great importance. This is also beneficial to the layout implementation process, and avoids the need of a post-procesing step over this values.

One of the objectives for the optimization of both OTA's topologies Miller and RFC is the minimization of the silicon area from the area calculation of each transistor that is part of the OTA. Therefore the area of each transistor was minimized to reduce the

area of the entire circuit. It should be noted that other features were considered, such as ensuring a minimum differential mode gain of 60dB and a GBW of 10MHz for the MIller OTA and greater than 100KHz for the RFC OTA.

Finally, is guaranteed that the operation point of each transistor is whithin the strong inversion region, this to ensure that the circuits are not affected by PVT variations. Subsequently, the layout of an OTA Miller was carried out, then post-layout simulations were performed and through the analysis of these results is demonstrated that the extracted circuit is also robust to PVT variations.

6.2. Future work

- Use algorithms that are capable of optimizing many objectives, in order to minimize or maximize all the characteristics of an OTA.
- Dimension more complex integrated circuits from the design variables by applying metaheuristics with a greater number of objectives.
- Implement and improve constraints handling to guide the algorithm to better solutions.
- Improve the processing time and memory space of the implemented algorithms

Bibliography

[1] M. F. Barros, J. M. Guilherme, and N. C. Horta, *Analog circuits and systems optimization based on evolutionary computation techniques*, vol. 294. Springer, 2010.

[2] B. Razavi, *Design of analog CMOS integrated circuits*. Tsinghua University Press Ltd., 2005.

[3] E.-G. Talbi, *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons, 2009.

[4] F. Maloberti, *Analog design for CMOS VLSI systems*, vol. 646. Springer Science & Business Media, 2006.

[5] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.

[6] P. E. Allen and D. R. Holberg, *CMOS analog circuit design*. Oxford university press, 1987.

[7] B. Liu, G. Gielen, and F. V. Fernández, "Automated design of analog and high-frequency circuits," in *A computational intelligence approach Springer*, Springer, 2014.

[8] R. Martins, N. Lourenço, N. Horta, J. Yin, P.-I. Mak, and R. P. Martins, "Many-objective sizing optimization of a class-c/d vco for ultralow-power iot and ultralow-phase-noise cellular applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, pp. 69–82, 2019.

[9] B. P. De, K. B. Maji, R. Kar, D. Mandal, and S. P. Ghoshal, "Design of optimal cmos analog amplifier circuits using a hybrid evolutionary optimization technique," *Journal of Circuits, Systems and Computers*, vol. 27, no. 02, p. 1850029, 2018.

[10] I. Guerra-Gómez, E. Tlelo-Cuautle, and G. Luis, "Richardson extrapolation-based sensitivity analysis in the multi-objective optimization of analog circuits," *Applied Mathematics and Computation*, vol. 222, pp. 167–176, 2013.

[11] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.

[12] F. Glover, M. Laguna, E. Taillard, and D. de Werra, *Tabu search*. Springer, 1993.

[13] M. Dorigo and M. Birattari, *Ant colony optimization*. Springer, 2010.

[14] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[15] K.-L. Du and M. Swamy, "Search and optimization by metaheuristics," *Techniques and Algorithms Inspired by Nature; Birkhauser: Basel, Switzerland*, 2016.

[16] B. Melián, J. A. M. Pérez, and J. M. M. Vega, "Metaheurísticas: Una visión global," *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, vol. 7, no. 19, p. 0, 2003.

[17] T. A. Feo and M. G. Resende, "Greedy randomized adaptive search procedures," *Journal of global optimization*, vol. 6, no. 2, pp. 109–133, 1995.

[18] K. Maji, R. Kar, D. Mandal, and S. Ghoshal, "An evolutionary approach based design automation of low power cmos two-stage comparator and folded cascode ota," *AEU-International Journal of Electronics and Communications*, vol. 70, no. 4, pp. 398–408, 2016.

[19] T. Liao and L. Zhang, "Efficient parasitic-aware hybrid sizing methodology for analog and rf integrated circuits," *Integration*, vol. 62, pp. 301–313, 2018.

[20] J. Yu and Z. Mao, "A design method in cmos operational amplifier optimization based on adaptive genetic algorithm," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 548–558, 2009.

[21] A. C. Sanabria-Borbón, E. Tlelo-Cuautle, and L. G. de la Fraga, "Optimal sizing of amplifiers by evolutionary algorithms with integer encoding and

$$g\_m/i\_d$$

design method," in *NEO 2016*, pp. 263–279, Springer, 2018.

[22] I. Guerra-Gómez, E. Tlelo-Cuautle, T. McConaghy, and G. Gielen, "Decomposition-based multi-objective optimization of second-generation current conveyors," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 220–223, IEEE, 2009.

[23] C. C. Enz and E. A. Vittoz, *Charge-based MOS transistor modeling: the EKV model for low-power and RF IC design*. John Wiley & Sons, 2006.

[24] Y. Cheng and C. Hu, *MOSFET modeling & BSIM3 user's guide*. Springer Science & Business Media, 1999.

[25] B. Razavi, *Fundamentals of microelectronics*. Wiley, 2008.

[26] S. Mallick, R. Kar, S. Ghoshal, and D. Mandal, "Optimal sizing and design of cmos analogue amplifier circuits using craziness-based particle swarm optimization," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 5, pp. 943–966, 2016.

[27] A. S. Khade, V. Vyas, and M. Sutaone, "Performance enhancement of advanced recycling folded cascode operational transconductance amplifier using an unbalanced biased input stage," *Integration*, 2019.

[28] S. T. Mathew and R. M. S. Laali, "Resourceful design methodology of a low power two stage cmos based operational transconductance amplifier," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 3699–3703, IEEE, 2016.

[29] G. Yosefi, "The high recycling folded cascode (hrfc): A general enhancement of the recycling folded cascode operational amplifier," *Microelectronics Journal*, 2019.

[30] A. S. Sedra and K. C. Smith, *Microelectronic circuits*. New York: Oxford University Press, 1998.

[31] M. P. Garde, A. Lopez-Martin, R. G. Carvajal, and J. Ramírez-Angulo, "Super class-ab recycling folded cascode ota," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2614–2623, 2018.

[32] R. S. Assaad and J. Silva-Martinez, "The recycling folded cascode: A general enhancement of the folded cascode amplifier," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 9, pp. 2535–2542, 2009.

[33] J. Brownlee, *Clever algorithms: nature-inspired programming recipes*. Jason Brownlee, 2011.

[34] A. Kaveh, *Advances in metaheuristic algorithms for optimal design of structures*. Springer, 2014.

[35] M. E. H. Pedersen, "Good parameters for particle swarm optimization," *Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001*, 2010.

[36] M. Köppen, G. Schaefer, and A. Abraham, *Intelligent Computational Optimization in Engineering: Techniques & Applications*, vol. 366. Springer Science & Business Media, 2011.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[38] L. G. de la Fraga and E. Tlelo-Cuautle, "Optimizing the maximum lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1503–1515, 2014.

[39] "Particle swarm optimization in MATLAB," Last time checked: May 22nd, 2019. `http://yarpiz.com/50/ypea102-particle-swarm-optimization`.

[40] J. C. F. Cabrera and C. A. C. Coello, "Handling constraints in particle swarm optimization using a small population size," in *Mexican International Conference on Artificial Intelligence*, pp. 41–51, Springer, 2007.