MULTI-DIMENSIONAL YIELD-AWARE OPTIMIZATION OF THE ANALOG AND HETEROGENEOUS CIRCUITS

by

Murat PAK

B.Sc., Electronics Engineering, Istanbul Technical University, 2008

M.Sc., Electrical and Electronics Engineering, Boğaziçi University, 2011

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2020

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude for my advisors Prof. Gunhan Dundar and Prof. Francisco V. Fernandez for their continuous support during my Ph.D. study and the related research, for their patience, motivation and immense knowledge. It was a great experience to have their mentorship during my Ph.D. study.

Besides my advisors, I would like to thank to Asst. Prof. Faik Baskaya from my thesis committee, for his comments on my progress reports, for his encouragement and all the other supports he has provided during my Ph.D. study.

My sincere thanks also go to my old colleagues from Tubitak YITAL team, supporting me while conducting this research.

I would like to especially thank to my dear wife Cigdem Demet Pak who has always supported me during my Ph.D. study. Thanks to her, the phrase “behind every successful man, there is a strong woman”, has simply been confirmed.

I am also thankful to my friends Mustafa Celik, Onur Ates and my cousin Can Canbaz for truly being the brothers I never had, for their inspirations and for the wonderful times we shared.

Last but not the least, I would like to thank to the rest of my family: my three years old daughter Ada Pak, my mother Hediye Pak, my father Halil Pak, my sister Aylin Kaymaz and my brother in law Dogan Kaymaz for their spiritual support throughout writing this thesis and for their existence in my life.

ABSTRACT

MULTI-DIMENSIONAL YIELD-AWARE OPTIMIZATION OF THE ANALOG AND HETEROGENEOUS CIRCUITS

Even though the design of digital circuits is well supported by several CAD tools, this is not the case for analog circuits and MEMS where the design is typically hand-crafted by expert designers. Several tools have been implemented for trade-off exploration in analog circuits; however, yield-optimization is a hot and vital topic considering that process variations have deteriorated along with the feature sizes scaling down; hence, physical variations originating from manufacturing process have a huge impact on yield. Therefore, efficient yield-aware optimization methodologies for analog ICs are needed.

MEMS design, on the other hand, requires a lot of expert knowledge, which implies long design times and increased cost due to this physical heterogeneity. The approach followed by the industry, based on composing separately designed sensors and read-out circuitry, has several issues such as inappropriate partitioning of system specifications or potential violation of system level constraints during the coupling process of the these devices. Hence, design methodologies which can obtain globally optimal MEMS by performing co-optimization of the sensor and the circuit are needed.

This study is mainly focused on developing and implementing novel and generic design methodologies for multi-objective yield-aware optimization of analog circuits and MEMS. A novel yield optimization technique has been proposed and compared with the existing approaches and has provided very promising results for yield-aware Pareto Front generation. Besides the work conducted for yield-aware optimization, co-optimization of MEMS and analog circuits has been performed for the first time by jointly optimizing a mechanical accelerometer sensor and an electronic read-out circuitry. The implemented yield-aware optimization techniques have been integrated into the co-optimization loop to enable yield-aware multi-objective optimization of MEMS.

ÖZET

ANALOG VE HETEROJEN DEVRELERİN ÇOK AMAÇLI VERİM OPTİMİZASYONU

Sayısal devre tasarımı çok sayıda CAD yazılımıyla destekleniyor olsa da analog devre ve MEMS tasarımları genellikle uzman tasarımcılar tarafından manuel olarak yapılmaktadır. Analog devre tasarımında transistör boyutlarındaki küçülmeyle birlikte proses sapmalarının devre performansına olan etkisinin artması, analog devre tasarımında verim optimizasyonunu çok önemli kılmaktadır. Bu sebeple analog devre tasarım sürecine dahil edilecek şekilde kullanılabilecek verim optimizasyonu teknikleri geliştirilmelidir.

Diğer taraftan MEMS tasarımı, mekanik ve elektronik blokları bir araya getiren fiziksel heterojenlik içermesi bakımından uzman tasarımcılar gerektiren bir süreçtir. Endüstri tarafından bugün izlenen temel yöntem mekanik sensör ile elektronik okuma devresinin ayrı ayrı tasarlandıktan sonra bir araya getirilerek sistem tasarımının gerçekleştirilmesidir. Bu yöntemin farkli blokların eşleştirilmesi sırasında ortaya çıkan, sistem spesifikasyonlarinin sensör ve okuma devresi arasında ideal olmayan şekilde paylaştırılması ve bu spesifikasyonlarin ihlal edilmesi gibi problemleri mevcuttur. Bu sebeple sensör ve okuma devresinin birlikte optimize edildiği yöntemlere ihtiyaç vardır.

Bu tez çalışması, temel olarak analog devre ve MEMS tasarımında kullanılabilecek çok amaçlı verim optimizasyonu tekniklerinin geliştirilmesi ve bunların örnek devre ile sistemler üzerinde test edilmesini kapsamaktadır. Tez çalışması kapsamında, yeni bir verim optimizasyonu yöntemi geliştirilmiş ve bu yöntem halihazırda bulunan tekniklerle karşılaştırılarak önerilen yöntemin üstünlüğü gösterilmiştir. Verim optimizasyonu çalışmalarına ek olarak, ilk kez, mekanik sensör ve elektronik okuma devresinin birlikte optimize edildiği bir yazılım geliştirilmiş ve bir MEMS ivme-ölçer sistemi üzerinde test edilmiştir. Geliştirilen verim optimizasyonu teknikleri de bu yazılıma dahil edilerek çok amaçlı ve gürbüz MEMS tasarımları geliştirilen yazılımlar aracılığıyla elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Acceleration
B	Neighborhood space
C_{gap}	Capacitive gap
d	Distance between the objective functions
E	Young's Modulus
f	Fitness function
F_t	Cut-off Frequency
g	Gravitational constant
I_b	Bias current
I_c	Collector current
k	Spring Constant
L	Length
m	Mass
P	Population
Q	Quality factor
t	Thickness
t_{ox}	Oxide thickness
V_a	Early Voltage
V_b	Base voltage
V_{th}	Threshold voltage
W	Width
w_r	Resonant Frequency
Y	Yield
Ω	Design variable search space
λ	Weight vector matrix

LIST OF ACRONYMS/ABBREVIATIONS

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
CD	Critical Dimension
CMOS	Complementary Metal Oxide Semiconductor
CW	Cloud Width
C/V	Capacitance to Voltage
DAC	Digital to Analog Converter
FOM	Figure of Merit
GBW	Gain Bandwidth Product
HDL	Hardware Description Language
IBY	Individual Based Yield
IC	Integrated Circuit
IGD	Inverted Generational Distance
LDS	Low Discrepancy Sequence
LHS	Latin Hypercube Sampling
LNA	Low Noise Amplifier
LM	Lebesgue Measure
MC	Monte Carlo
MEMS	Micro Electro Mechanical System
MIM	Metal Insulator Metal
MOEA	Multi Objective Evolutionary Algorithm
MOEA/D	Multi Objective Evolutionary Algorithm with Decomposition
MOGA	Multi Objective Genetic Algorithm
MOP	Multi Objective Optimization Problem
NF	Noise Figure
NSGA	Non-dominated Sorting Genetic Algorithm
OPC	Optical Proximity Correction
PF	Pareto Front
PLL	Phase Locked Loop

QMC	Quasi Monte Carlo
QQ	Quasi Quasi
RSB	Response Surface Based
SiGe	Silicon Germanium
SoC	System on Chip
SPEA	Strength Pareto Evolutionary Algorithm
SW	Software
S/H	Sample and Hold
TIA	Transimpedance Amplifier
VEEA	Vector Evaluated Evolutionary Algorithm
VGA	Variable Gain Amplifier
VLSI	Very Large Scaled Integrated
WCD	Worst Case Distance
WCPF	Worst Case Pareto Front

1. INTRODUCTION

1.1. An Introduction to the Design Optimization of Analog Circuits and MEMS and the Motivation of the Dissertation

Communication with the nature, just like the nature itself is analog. With the progress in the capabilities of computers, as well as digital processing techniques, where discrete signals are used, it was initially assumed that the importance of analog circuit design was going to diminish; however, analog circuits became even more vital due to the need for interface circuits between the real analog world and the digital processing circuitry [1].

Even though there is a trend to replace analog circuitry by digital whenever possible, there are three types of circuits that will always remain analog, for both application specific integrated circuits (ASICs) and systems on chip (SoCs). The first type corresponds to the circuits at the input side of a system, where signals coming from a sensor, microphone, antenna, etc. are first sensed and then amplified and/or filtered up to a certain level which can be used for digitization. Some examples are low noise amplifiers (LNA), filters, oscillators, and variable gain amplifiers (VGA). The second type of analog circuits are the ones used at the output of a system. The signal converted back to analog (from digital) must be amplified to be able to drive a load at the system output. Typical examples for these loads are antennas and loudspeakers. Some examples of this type of analog circuits are buffers, filters, mixers, and oscillators. The last type of analog circuits are the circuits used in mixed signal systems where analog and digital signals coexist. In mixed signal circuits/systems, the analog and digital parts are integrated with each other. Typical examples of the analog circuits used in mixed signal design are sample-and-hold (S/H) circuits, phase locked loops (PLL), analog to digital converters (ADC), and digital to analog converters (DAC). Moreover, any type of circuit requires stable reference signals for operating properly and this is provided by analog circuits such as voltage and current reference circuits as well as oscillators [2]. Due to the reasons given above, the analog circuits are expected to continue being widely used in the near future.

Another technology which is frequently used nowadays is micro-electromechanical systems (MEMS). MEMS is a process technology which includes both the mechanical (sensors and actuators) and electrical components in a system by combining silicon-based microelectronics with micromachining technology, hence carrying heterogeneous (mechanical and electrical) signals in the same system. An example where micro-electromechanical accelerometer sensors are integrated on a single chip (SoC) with the analog and digital components is given in Figure 1.1. below.

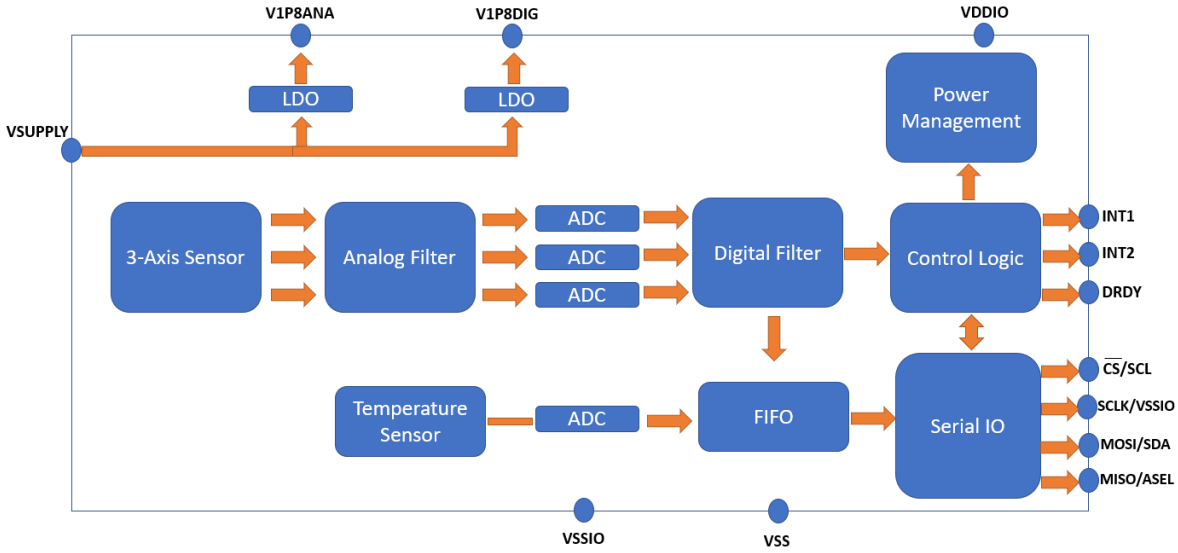

Figure 1.1. ADXL355 3-Axis MEMS Accelerometer by Analog Devices [3]

Today, MEMS are used in several applications such as automotive, defense, communication, medical, etc. Some of the current MEMS devices include pressure sensors, accelerometers, flow sensors, mirror devices, gas sensors, micromotors, and microgears. Hence, just like the analog circuits, MEMS devices are widely used in today's technology and will continue being in the market in the following decades.

Growing requirements, especially for single chip system designs, together with the common trends towards smaller feature sizes and higher scales of integration have brought about new dimensions in the complexity of microelectronic system design. While the design of digital circuits is supported by mature computer-aided design (CAD) tools, analog CAD tools [1] as well as CAD tools for MEMS [4] are much more scarce. In digital circuit design, structured abstractions and hierarchy are used to generate complex systems with large numbers of devices. For example, digital circuits can be easily designed by using synthesis tools which translate the hardware description languages (HDL) to a gate-level

architecture. In contrast, much of the design of analog circuits, as well as MEMS are still hand-crafted by expert designers. Efficient design methodologies supported by analog and micro-electromechanical CAD programs are needed not only to improve the design performance, but also to speed up the design process. CAD tools can improve the design process in several ways such as reducing the design times, simplifying the design process, preventing design errors, reducing manufacturing and design costs, improving the device/circuit performance and improving the yield. CAD tools, which are components of the design cycle, can be classified as simulation, modelling, layout, analysis, synthesis, and verification [5]. This dissertation is focused on implementing efficient and accurate synthesis tools for analog circuits, as well as heterogeneous systems including analog circuits and MEMS sensor or actuators.

Synthesis tools are in most cases design optimization tools, where certain design variables of a circuit (e.g., transistor dimensions and bias current for an analog circuit) or a device (e.g., finger widths of beam structures and thickness of the mass for a MEMS sensor) are optimized to obtain optimum performance, satisfying a set of specifications. Design optimization tools, together with yield-aware optimization approaches, can also enable robust optimization of the circuits or the systems.

As the complexity of the analog circuits and the MEMS grow, developing efficient CAD tools for design optimization becomes an urgent requirement. Complex trade-off exploration in MEMS design, for instance, requires a lot of expert knowledge, which implies long design times and increased cost especially due to the physical heterogeneity of the MEMS where a mechanical sensor output is usually read-out by an analog circuit. Consequently, trial and error approaches are still widely used for MEMS design. Design automation, on the other hand, dramatically shortens the design time and enables optimal designs, which even an expert designer cannot easily achieve. Among the automated design methodologies, optimization-based ones are especially promising for complex mixed-domain systems such as MEMS [6].

Design of MEMS comprises different domains like the mechanical MEMS sensor and the electronic read-out circuitry for signal conditioning; therefore, being a challenging problem. Industry has traditionally followed ad-hoc approaches based on the separate design

of the MEMS sensor and the circuitry by different engineering teams and, then, combining these two parts. This can be either achieved by using a pre-designed MEMS sensor and then, designing the electronic circuitry accordingly, or by composing an optimal sensor design with an optimal circuit design (which are designed in parallel or obtained from a library) in order to obtain a so-called optimal system performance. However, combining these separately designed blocks is problematic due to several reasons. A first problem is that inappropriate partitioning of system specifications among the mechanical and electronic parts might lead to non-optimal system performance.

Another problem that can appear during the composition of independently designed blocks is that some system level constraints may be violated. For example, the sensor itself can be designed to satisfy a certain measurement range, and its composition with an inappropriate circuit design may lead to a measurement range of the system which does not satisfy the system specifications due to an unsuitable level of amplification. As a result, the coupling of the separately designed mechanical and electronic devices is a challenge. Hence, design methodologies that simultaneously consider both the mechanical and the electronic parts in order to develop a multi-domain MEMS synthesizer is needed.

Several optimization-based approaches have been developed for the synthesis of MEMS sensors. Some of them are focused on the optimization of the MEMS device by either performing device level simulations of the MEMS sensors within an optimization loop [7,8] or using accurate analytical models to speed up the synthesis process [9–11]. Some other efforts [12–14] have been devoted to multi-domain optimization of MEMS, mostly being inspired by VLSI hierarchical design methodologies, that aim at decomposing the mixed-domain design problem into smaller optimization sub-problems, like the sensor itself and the input stage electronics. The approach in [12] focuses on developing a design methodology for the MEMS sensor where a mixed-domain circuit simulation is performed to obtain transient simulation results of the complete MEMS. However, the synthesis results focus on the MEMS sensor only instead of the whole system. In [13], the MEMS sensor and the read-out circuit are designed at the behavioral level to enable their joint simulation. The design parameters at the circuit level are only a few high-level parameters such as the modulation voltage. None of the circuit level design parameters, like the transistor sizes or the values of the passive devices, have been considered. Hence, only a high-level

architectural description, dedicated to a sigma-delta control loop, is defined, instead of developing a device-level solution that is applicable to any MEMS. In [14], on the other hand, a top-down design methodology, which is based on optimizing the sensor first and then, the circuitry using the optimal sensor design has been reported. This type of methodology has several drawbacks, such as the inappropriate partitioning of the system specifications or the potential the violation of system level constraints during the coupling process of the mechanical and electronic devices. Therefore, to the best of our knowledge, there was no generic design methodology that can obtain globally optimum MEMS until our study in [6] where the sensor and circuit co-optimization of the MEMS was performed for the first time.

The scenario is significantly different in analog circuit optimization, which has already received quite a lot of interest and effort. An overview of some of the analog synthesis tools in the literature can be found in [15]. On the other hand, robustness of the designs, and hence yield optimization, is an obviously crucial and challenging problem in analog circuit optimization. Especially in deep-submicron technologies, process tolerances have deteriorated as feature sizes scale down. As long as it is not possible to remove the random physical variations coming from the manufacturing environment, it is important to efficiently model these variations and incorporate them into the design process. For that reason, designing robust circuits, which are less sensitive to variations, has become a must in IC design [16]. For analog ICs, besides the robustness concept, trade-offs between different circuit performances complicate the design of the circuit; hence, increased design times are required.

Figure 1.2 illustrates the robustness problem. The region bounded by the dashed lines represents the acceptable region of a circuit. The two upper solutions show the nominal performance values while the lower left ones illustrate the electrical performance deviations on a couple of circuit instances coming from the process variations, which are unavoidable [17].

As shown in Figure 1.2, the green solution does not stay in the acceptable region after the inclusion of the process variations, while the blue one is still inside the acceptable region making it a robust solution. Hence, for synthesizing robust analog circuits,

performance variations must be taken into account and integrated into the optimization flow. In order to address this problem, several yield-aware design automation tools have been developed [17]. The details of these algorithms will be given in Chapter 4 and their implementation in analog circuit optimization will be presented in Chapter 5.

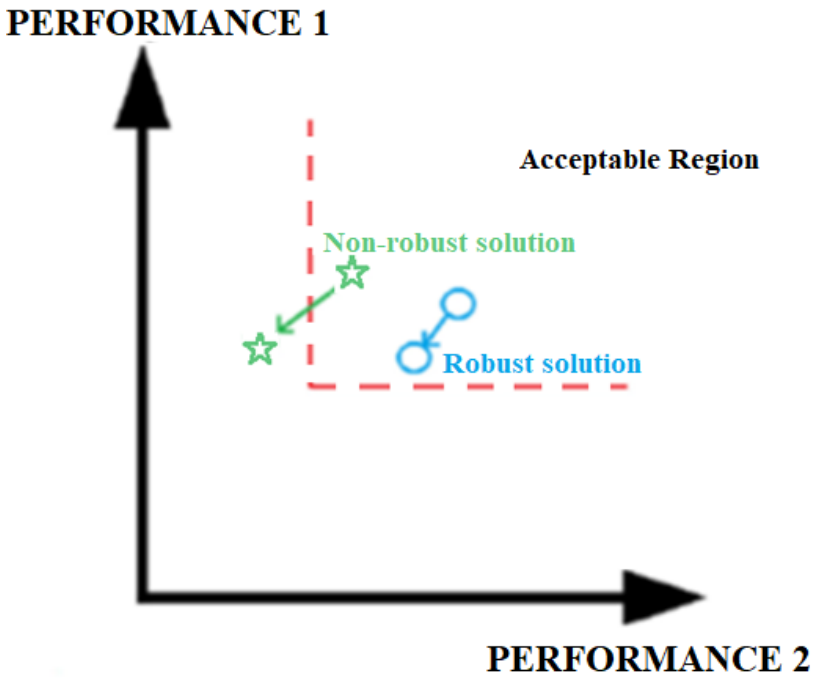

Figure 1.2. Illustration of Robust and Non-robust Designs. [17]

Process variations play a very important role in the robustness of mixed-domain MEMS (mechanical sensor/actuator and analog read-out circuitry) as well; and, should also be considered during the optimization of the MEMS to avoid re-spins in the fabrication. For the analog circuits, as mentioned above, it is a well-known fact that process variations have deteriorated along with the feature sizes scaling down; hence, the random physical variations originating from the manufacturing process have a much bigger impact on yield. The device geometry for the mechanical sensors, on the other hand, might be relatively large; however, even small variations on certain device parameters, such as the capacitive gap between the electrodes of an accelerometer, have a huge impact on the device performance. For that reason, considering these variations during the design process is a must in order to guarantee robust system designs for MEMS. Besides our study in [4], which carries out the multi-objective (which means more than one objective is considered as the performance specification to be optimized) yield-aware optimization of a MEMS accelerometer system,

the robustness of MEMS has also been studied in [18]. This work [18] is also focused on the yield-aware optimization of a capacitive MEMS accelerometer; however, with a different methodology based on design centering with a worst-case distance (WCD) approach. Our approach, on the other hand, focuses on a different methodology where the yield-aware Pareto Fronts with hundreds of robust design points are generated. More details of our approach for the multi-objective yield-aware optimization of MEMS will be presented in Chapter 6.

1.2. The Novel Contributions

During this study, several novel contributions have been realized, resulting in the publication of three conference and three journal papers. The overview of the novel contributions is listed below:

- A novel methodology for the integration of the yield-awareness to the optimization loop has been defined and implemented.
- A novel yield constraint technique, based on the span of the performance variability, has been implemented and this technique has been compared with the other existing methodologies using both analog circuit and MEMS synthesis.
- Multi-objective nominal optimization of a MEMS sensor has been performed by implementing an analytical model-based approach for the first time.
- Multi-objective yield-aware optimization of a MEMS sensor has been performed by implementing an analytical model-based approach for the first time.
- Co-optimization of the mixed-domain MEMS system has been implemented by jointly optimizing the mechanical sensor and the analog read-out circuitry for the first time.

1.3. Outline of the Dissertation

The thesis document is structured as follows. Chapter 2 focuses on the optimization algorithm used in the dissertation. Initially, the motivation for the selection of the MOEA/D optimizer is given. Then, the details of the MOEA/D algorithm [19], used for optimization purposes, are explained. Finally, the evaluation of the MOEA/D algorithm is realized by performing optimizations of a folded-cascode amplifier with up to five performance

objectives. Several performance metrics are used for the evaluation of the optimizer and they are also explained in detail.

In Chapter 3, process variations in analog integrated circuits, mechanical sensors, and their impacts on the system performance are discussed. The chapter starts with an introduction to the variability concept, where physical and electrical variations are explained. It is followed by the definition of intra-die and inter-die variations and the impact of these variations on the device, circuit, and system performances. In the second part of the chapter, different methodologies used for statistical analysis of the devices and the circuits are reviewed, and the reasons for selecting a Monte Carlo based methodology are clarified. Finally, statistical analysis of analog components such as CMOS and bipolar transistors, passive devices and a capacitive MEMS accelerometer sensor are performed.

Chapter 4 starts with an introduction to the concept of yield and a review of different yield-aware optimization methodologies. The definition of process variations as a yield constraint is followed by the integration of this constraint in the main optimizer. A novel technique called CW is implemented and compared with several existing methodologies such as the worst-case solution optimization (denoted as WCPF, worst case Pareto front) and direct use of the yield constraint based on the acceptance region (denoted as IBY, individual based yield) techniques. For the integration of the yield into the optimization algorithm, a new technique called “transmission factor” is implemented. The details of this technique are also explained in the chapter.

In Chapter 5, the yield-aware optimization of a fully differential folded-cascode amplifier is performed. First, the topology of the amplifier, the design variables used, and the objectives to optimize are given. Later, three different yield-aware optimization methodologies mentioned in Chapter 4 are implemented and both two- and three-dimensional yield-aware optimizations of the fully differential folded-cascode amplifier are carried out. The yield-aware optimization techniques are compared based on the final robust design points.

In Chapter 6, the optimization of a capacitive MEMS accelerometer system is performed. The chapter starts with the details of the capacitive accelerometer sensor and the

system. The topology and the working principles of the mixed-domain system (mechanical sensor and analog read-out circuitry) are presented. It is followed by the selection of the design variables and objectives used for both sensor and system optimizations. The optimizations performed in this chapter start with the nominal (without considering the yield) optimization of the capacitive MEMS sensor. The details of the highly accurate analytical model used for the sensor is explained and a two-dimensional optimization of the MEMS sensor is carried out. Later, the two-dimensional nominal optimization of the capacitive MEMS accelerometer system is performed. Finally, the yield-aware optimization tool for the robust MEMS accelerometer system synthesis is implemented for two different yield-aware optimization techniques (CW and IBY) in order to run two- and three-dimensional yield-aware optimization of the MEMS accelerometer. The chapter is concluded by comparing the results obtained for different yield-aware optimization techniques.

The last chapter concludes the dissertation by drawing some conclusions and discussing possible future improvements.

2. THE MULTI-OBJECTIVE SYNTHESIS TOOL

This chapter starts with the motivation for the selection of the MOEA/D algorithm for both analog circuit and MEMS optimizations. The multi-objective optimization concept, together with the details of the selected MOEA/D algorithm, is also explained. The chapter finally includes the evaluation of the optimization algorithm by running up to five-dimensional optimization of a folded-cascode amplifier.

2.1. Motivation for the Selection of the MOEA/D Optimizer

Optimization in CAD is known as finding the optimal circuit and system performance by altering the design variables. For analog circuits, some typical examples of design variables are the width (W) and length (L) values of the transistors, which are optimized to obtain the best specifications such as phase margin, power, gain etc. The geometry of the beam and mass structures for the MEMS sensors and actuators are typical design parameters for MEMS, and are optimized to obtain the best sensor/actuator performances such as noise, measurement range, etc. For the optimization of analog circuits, complex device models are used, and the optimization problem is a nonlinear and nonconvex problem where locally optimal design points can be encountered. Moreover, the lack of precise analytical models or any other accurate techniques for the performance functions typically enforces time-consuming simulations in the optimization process. For MEMS sensors and actuators, on the other hand, highly accurate analytical models can be generated and used in the optimization loop for faster performance evaluations.

There are several optimization methodologies proposed in CAD tools for analog synthesis, some of which are promising for MEMS optimization as well. Compared to traditional optimization techniques, evolutionary algorithms are more robust methodologies and can obtain a better balance between the efficiency (the optimization speed) and efficacy (quality of the final solutions) for many different problems [20]. This property makes them a perfect candidate for complex problems such as analog circuit and MEMS optimization.

There are several components of evolutionary algorithms. The structure of an Evolutionary Algorithm is given in Figure 2.1. The evaluation functions define how good that individual for the performances to be optimized is. Population, on the other hand, is a set of individuals which generates the solution set. Population size is an important parameter in optimization algorithms, and it means the number of individuals in a population. The survivor selection is the mechanism which lets the better individuals to be transferred to the next generation as the new parents. Together with the survivor selection mechanism, parent selection is responsible for achieving quality improvements. In evolutionary computation, parent selection is typically probabilistic. High quality individuals get a higher chance to become parents than those with low quality. Another important component of evolutionary algorithms is the variation operators which creates new individuals from old ones. Variation operators in evolutionary computation are the mutation and recombination operators. Once a mutation operator for instance, is applied to one genotype, it delivers a slightly modified mutant (known as the child). Recombination operator, on the other hand, is used to combine the genetic information of two parents to generate new offspring. The last component is the replacement, which is also known as the survivor mechanism. The aim of the survivor selection is to distinguish among the individuals based on their quality.

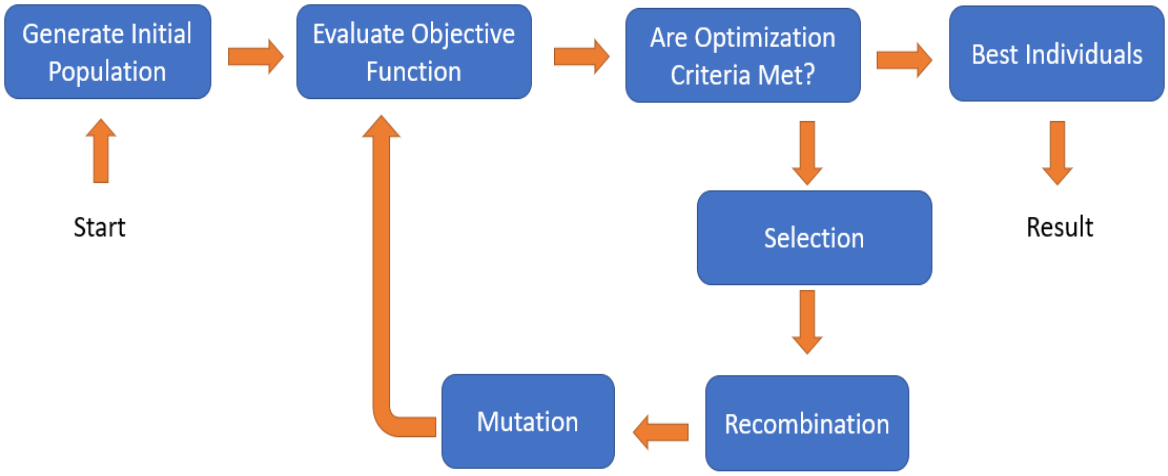

Figure 2.1. Structure of an Evolutionary Algorithm [21]

As it can be seen, an initial population is generated randomly, and then according to the results obtained from the evaluation of the objective functions, the optimization algorithm alters the design variables by using evolutionary operators (which are genetic operators in the implemented tool). The optimization loop continues until a stopping

criterion (i.e., number of the iterations allowed or the limits defining the improvements in the fitness values are reached) is met.

In the last decades, the interest in "Evolutionary Multi-objective Optimization" has increased dramatically. Since these objectives may be conflictive between each other, it is not possible to find a single and best solution which optimizes all the objectives. Optimal performance for a particular objective for instance, often implies low performance in one or more of the other objectives. Evolutionary algorithms explore a set of possible solutions simultaneously instead of a single solution. Pareto optimal points in the objective space are those points where an improvement in a particular objective can only be satisfied with the performance deterioration of at least one other objective. This is known as the trade-off between the objectives [22].

The general flow of multi-objective evolutionary algorithms (MOEA) is given in Figure 2.2. A population of individuals is maintained during the search process of the MOEAs and at time t given as $P(t) = \{p_1(t), ..., p_N(t)\}$. Each $p_i(t) \in P(t)$ represents a potential solution to the problem. The probabilistic operators are used in order to generate a new and better population $P(t+1)$ [20].

```
t = 0
Initialize P(t)
Evaluate P(t)
While (termination condition !=1) {
        P'(t) = Reproduction of P(t)
        Evaluate P'(t) U P(t)
        P'(t+1) = Select P'(t) U P(t)
        t = t+1
}
```

Figure 2.2. The General Flow of a MOEA [20]

There are several approaches developed for multi-objective optimization using evolutionary algorithms. Some examples are the vector evaluated evolutionary algorithm (VEEA) [23], the multi-objective genetic algorithm (MOGA) [24], the non-dominated sorting genetic algorithm (NSGA) [25] and NSGA-II [26], and the strength Pareto evolutionary algorithm (SPEA2) [27].

As the main focus of this dissertation is implementing yield-aware optimization methodologies for both MEMS and analog circuits, powerful optimizers are needed. The performance comparison of MOEA/D, multi-objective evolutionary algorithm with decomposition, with other existing multi-objective evolutionary algorithms, such as NSGA-II, can be found in [5], which shows the superiority of MOEA/D. Moreover, it has been reported in [28] that MOEA/D is very powerful compared to several other techniques, for both analog circuit synthesis and mathematical benchmark problems. Including yield-awareness into the optimization loop, the synthesis process gets more complicated, since a higher number of objective functions and constraints are required [29]. In [30], it is also shown that MOEA/D is suitable for multi-objective yield-aware optimization. Considering all these previous studies, MOEA/D has been selected as the main optimizer for all nominal and yield-aware syntheses of both the analog circuits and the MEMS.

2.2. Details of the MOEA/D Optimizer

MOEA/D optimizer is used in both nominal and yield-aware synthesis tools in order to solve complex multi-objective optimization problems. A multi-objective optimization problem (MOP) can be stated as follows:

$$min \ \{f_1(x), \ldots, f_m(\mathrm{x})\}, x \in \Omega \tag{2.1}$$

where $x = (x_1, \ldots x_n)$ is the design variable vector and $f_i(x)$ are the objective functions. Ω is the design variable space.

Considering f_i for every $i \in \{1, \ldots, m\}$ as continuous functions, the MOP given in Eq. (2.1) is a continuous MOP. Very often, as the objectives given in Eq. (2.1) contradict each other, no design variable set in Ω can optimize all the objectives simultaneously. The Pareto

optimality is known as where no objective value can be any better without making at least one other objective value worse [20].

A solution x is said to dominate solution y if and only if $f_i(x) \leq f_i(y)$ for every $i \in \{1, \dots, m\}$ and $f_j(x) < f_j(y)$ for at least one index $j \in \{1, \dots, m\}$. A point $x^* \in \Omega$ is Pareto-optimal if there is no point $x \in \Omega$ such that $f(x)$ dominates $f(x^*)$. $f(x^*)$ is Pareto-optimal objective vector and the set of all Pareto-optimal objective vectors is known as the Pareto Front (*PF*).

The objective functions of a multi-objective optimization problem are the performances such as gain for analog circuits or system noise for MEMS, and these performance values can be the optimal solutions of a scalar optimization problem. It means that the *PF* approximation can be decomposed into a number of single-objective optimization subproblems. In order to define a single fitness function for the multi-objective optimization problem, the aggregation functions can be constructed by implementing several decomposition methodologies such as the weighted sum approach and the Chebyshev approach [20]. During the decomposition of the subproblems, weight vectors are assigned to the different subproblems in order to enable the calculation of an approximate PF [5].

In order to explain how MOEA/D works, let $\lambda^1, \dots, \lambda^N$ be a set of evenly spread weight vectors, for N being the number of subproblems, f_i's the objective functions, and z the vector storing the ideal point (best value) of each objective. The generation of the *PF* is, as mentioned before, decomposed into N scalar optimization subproblems by using the Chebyshev approach. The decomposed objective function g of the k^{th} subproblem, which is known as the fitness function to be minimized is:

$$g^{te}(x|\lambda^k, z^*) = \max_{i \leq 1 \leq m} \{\lambda_i^k |f_i(x) - z_i|\} \tag{2.2}$$

for $\lambda^k = \left(\lambda_1^k, \dots, \lambda_m^k\right)^T$.

The initialization phase is performed by randomly selecting the initial design variables. After that, the Euclidean distances between the weight vectors are calculated in order to find the T closest weight vectors to the i^{th} weight vector which are defined as the

neighbors $B(i) = \{i_1, \dots, i_T\}$. T is called niche, which is the predefined value for the number of neighbors. The weight vectors and the neighborhood concept are illustrated in Figure 2.3.

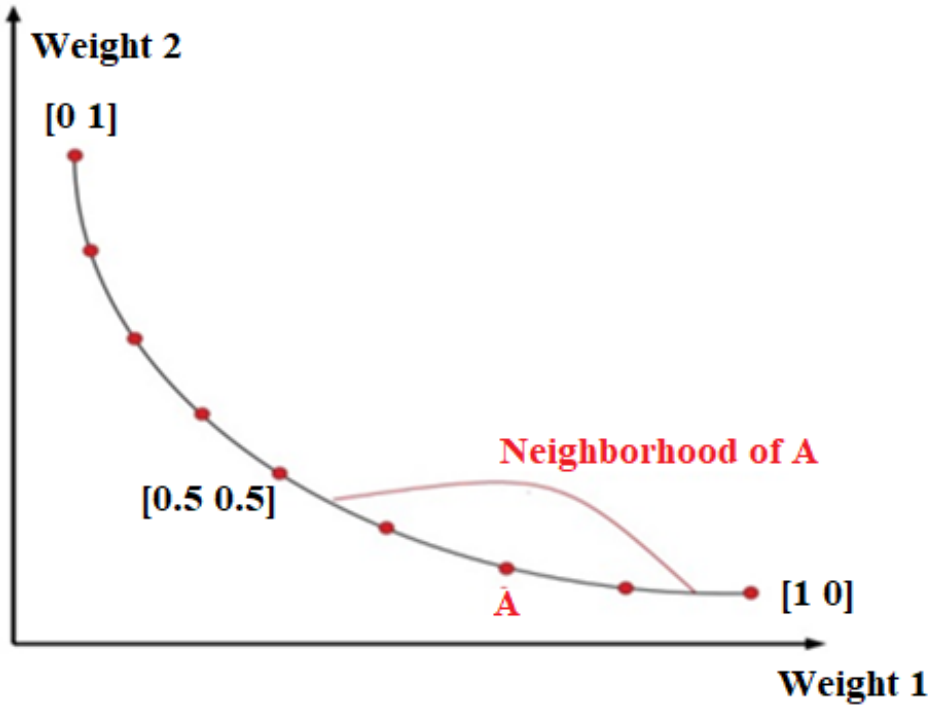

Figure 2.3. The Illustration of the Neighborhood Concept and Distribution of the Weight Vectors [17]

The index i will be the first index of $B(i)$ as it is the closest weight vector (itself already). The following *T-1* indexes are determined by the Euclidean distance to i^{th} vector and if an index *j* is a member of $B(i)$, then it can be told that *j* is a neighbor of *i* [5].

During the initialization step, the values of $z = (z_1, \dots, z_m)^T$ for each objective is also set to infinity (implemented by selecting a very high value) for a minimization problem.

After the initialization step, an initial population (the set of solutions) is obtained by evaluating the randomly selected design variables. After that, the optimization algorithm starts to run for a loop of *M* iterations. At each iteration, the solutions and the z values are updated.

In order to update the solutions with improved ones, first, two randomly selected indices of $B(i)$ are used to generate a new solution y' by using evolutionary operators. Later,

the new solution is used to calculate the values of the objective functions in order to update the z values. Later, $g^{te}(y'|\lambda^i, z) \leq g^{te}(x^i|\lambda^i, z)$ is checked for each subproblem in $B(i)$ and y' is set as the new solution for all neighbor solutions satisfying. This is done for all the subproblems. It should be noted that different indices of $B(i)$ have different neighbors, so the information is varied in a parallel (fast) and effective way.

After the stopping criterion is met, the final *PF* for the multi-objective problem optimized is obtained.

2.3. Evaluation of the MOEA/D Algorithm for High Dimensional Optimization

A folded-cascode amplifier has been designed in order to show that the synthesis performance does not meaningfully degrade when the number of dimensions is increased [29]. The folded-cascode amplifier has been optimized using 3, 4, and 5 objectives, using the TSMC 180nm CMOS technology.

2.3.1. A Practical Design for the Evaluation of MOEA/D: Folded-Cascode Amplifier

The folded-cascode amplifier topology given in Figure 2.4 has been used as a practical design example in order to check the MOEA/D optimizer performance for high dimensional, hence complex, circuit optimization problems.

In the amplifier circuit, there are 13 transistors and 1 current source whose values must be altered for optimum design. Hence, they are the design variables. Because of the design characteristics (i.e., symmetrical geometry of the current mirrors), some of the transistor dimensions are forced to be equal to each other. For the optimization of the folded-cascode amplifier, there are 10 different values of *W* and *L* to be optimized. The last design variable is the bias current value *Ib*. Table 2.1 shows design variables, while Table 2.2 shows the allowable ranges of the search space of the design variables or in other words, the minimum and maximum values that the design variables can take.

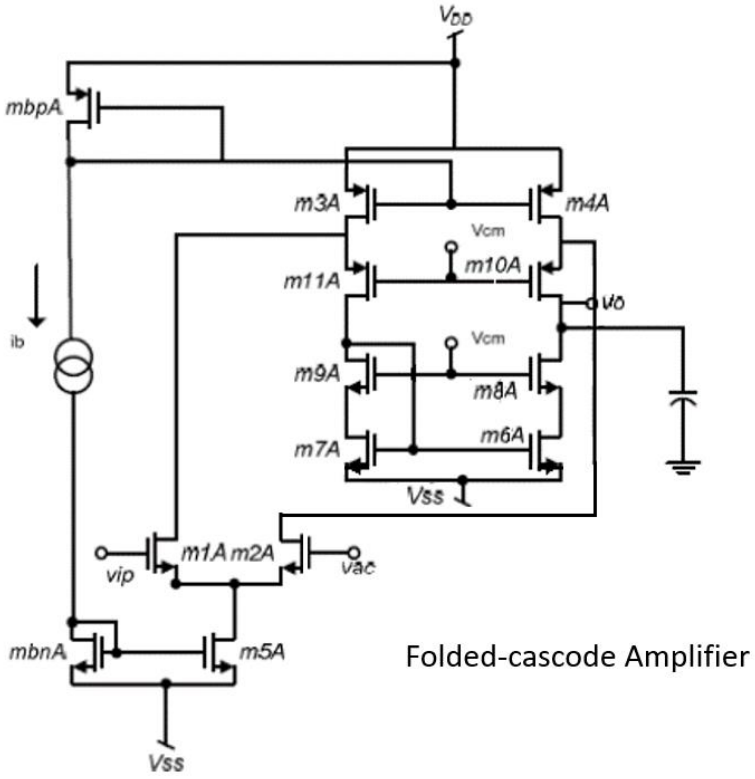

Figure 2.4. The Folded-Cascode Amplifier Topology [6]

Table 2.1. W and L variables to be optimized for the folded-cascode amplifier

Transistor	Width	Length
m1A	*W1*	*L1*
m2A	*W1*	*L1*
m3A	*W3*	*L3*
m4A	*W3*	*L3*
m5A	*W5*	*L5*
m6A	*W5*	*L5*
m7A	*W5*	*L5*
m8A	*W8*	*L8*
m9A	*W8*	*L8*
m10A	*W10*	*L10*
m11A	*W10*	*L10*
mbnA	*W3*	*L3*
mbpA	*W5*	*L5*

Table 2.2. Limit values of the design variables of the folded-cascode amplifier

Type of the design variable	Minimum Value	Maximum Value
W	240nm	800µm
L	180nm	10µm
Ib	0.5µA	2.5mA

The objectives and constraints used for optimizations with different number of dimensions are given in Table 2.3. The transistors are, as a constraint, forced to operate in the saturation region.

Table 2.3. Objectives and constraints for different dimensional optimizations of the folded-cascode amplifier

Performance	3D Synthesis	4D Synthesis	5D Synthesis
Gain	Objective #1	Objective #1	Objective #1
Gain * Bandwidth	Objective #2	Objective #2	Objective #2
Power	Objective #3	Objective #3	Objective #3
Area	Constraint	Objective #4	Objective #4
Phase Margin	Constraint	Constraint	Objective #5
Slew Rate	Constraint	Constraint	Constraint
Output Swing	Constraint	Constraint	Constraint
Operating Points	Constraint	Constraint	Constraint

During the optimizations, the population sizes have been selected as 100, 150, and 200 for three, four, and five dimensional optimizations respectively, with a stopping criterion of 100, 200, and 300 iterations. The niche has been selected as 30% of the population size for each optimization.

2.3.2. Performance Metrics used for Pareto Front Comparisons: Coverage Set, IGD and Schott's Spacing

The goal of this evaluation is to check if the algorithm can keep its performance for higher number of dimensions. To compare the performance of different Pareto Optimal Fronts, several performance metrics have been identified. Three of these metrics, which are IGD, Coverage Set and Schott's Spacing are used for the comparison of the different Pareto Fronts obtained in terms of the distribution quality, dominance, and the range of the solutions.

2.3.2.1. Coverage Set. Coverage set is a metric used for the dominance comparisons of two different Pareto Fronts. The percentage of the solution points on a certain Pareto Front dominated by the second Pareto Front is called the coverage set [31].

As an example, consider a Pareto front PF1 obtained using a certain methodology and another Pareto front PF2 obtained using a different methodology. If PF1 has 20 dominated points (20%, for a population size of 100) compared to PF2 and PF2 has 3 dominated points (3%, for a population size of 100) compared to PF1, it is expected that the second methodology has better optimization performance in terms of the dominance quality; hence, even if the first methodology may converge to the best *PF*, a longer synthesis time will be needed.

2.3.2.2. IGD Metric. An approximation to the true Pareto Front is typically generated by using the combination of the Pareto optimal solutions obtained from different runs or from different methodologies used for the optimization of the same problem. To be able to do that, initially, the Pareto Fronts obtained are merged and the dominated solution points in this merged PF are found out and eliminated. Hence, the points which have not been eliminated define the true Pareto approximation is used as a reference front for the calculations in several performance comparison metrics [5].

The inverted generational distance (IGD) metric [32] is a widely used Pareto Front comparison metric. Let P^* be a set of uniformly distributed points in the true Pareto Front and let A be an approximation to the PF. The *IGD* is defined as the distance from P^* to A:

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v,A)}{|P^*|} \tag{2.3}$$

where $d(v, A)$ is the minimum Euclidean distance between the points in A and v, where v refers to the solutions on the true Pareto Front. Hence, IGD is defined as the averaged sum of the minimum Euclidean distances of the solutions on a certain Pareto Front, to the solutions on the approximated true Pareto Front.

2.3.2.3. Schott's Spacing. Schott's Spacing metric is focused on finding how evenly the solutions are distributed on a Pareto Front and does not require a true Pareto Front to be calculated. It is an independent metric and can be generalized to different dimensions. Schott describes the following metric for spacing:

$$\sqrt{\frac{1}{n-1}\sum_{i+1} n.\left(\overline{d}-d_i\right)^2} \tag{2.4}$$

where the *n* is the total number of the solutions, *i, j=1,...,n* are the indices for certain solutions on the Pareto Front. Considering a two-dimensional optimization, as an example, f_1 and f_2 are the two objective functions, $d_i = min_j\left(\left|f_1^i(\vec{x})-f_1^j(\vec{x})\right| + \left|f_2^i(\vec{x})-f_2^j(\vec{x})\right|\right)$, and $\overline{d}$ is the mean distance of all d_i distances [33].

2.3.3. Projection Based Comparison of Optimizations with Different Dimensions

Obviously, it is not possible to compare two different Pareto Fronts which have different dimensions (which also means at least one different objective function). To achieve this goal, a methodology based on the projection of the Pareto Fronts to a lower dimension has been implemented.

For this purpose, the comparisons have been carried out by using N-dimensional Pareto Fronts as well as the N-dimensional projections of (N+1)-dimensional Pareto Fronts. For example, after the three-dimensional optimization of the folded-cascode amplifier, a three-dimensional Pareto Front with gain, gain-bandwidth product and power is obtained. For four-dimensional optimization on the other hand, there is an extra dimension which is the area. It is not possible to directly compare these fronts; hence, for the four-dimensional optimization, the Pareto Front is projected on a three-dimensional Pareto Front (which includes the gain, gain-bandwidth product and power objectives) where only the solutions with the area (fourth objective) performance satisfying the area constraint defined in the three-dimensional optimization are kept. Thus, fair comparison conditions with the same dimensional Pareto Fronts are created. Moreover, each synthesis has been run three times in order to have more statistically significant performance comparisons.

In order to generate an approximation to the true Pareto Fronts, 30 different optimizations have been run and the Pareto Fronts obtained have been combined. 30 has been selected as a practical number of the runs to generate the approximation to the true Pareto Front. Increasing that number further should ideally increase the accuracy of the obtained Pareto Front to the true Pareto Front.

The Coverage Set, IGD and Schott's Spacing metrics mentioned before have been used to compare the Pareto Fronts obtained for three-, four- and five-dimensional optimizations of the folded-cascode amplifier. Figure 2.5 shows a two-dimensional projection of different synthesis results to illustrate the fact that the solution space boundaries and the solution distributions of different objectives are similar to each other for different dimensional tests.

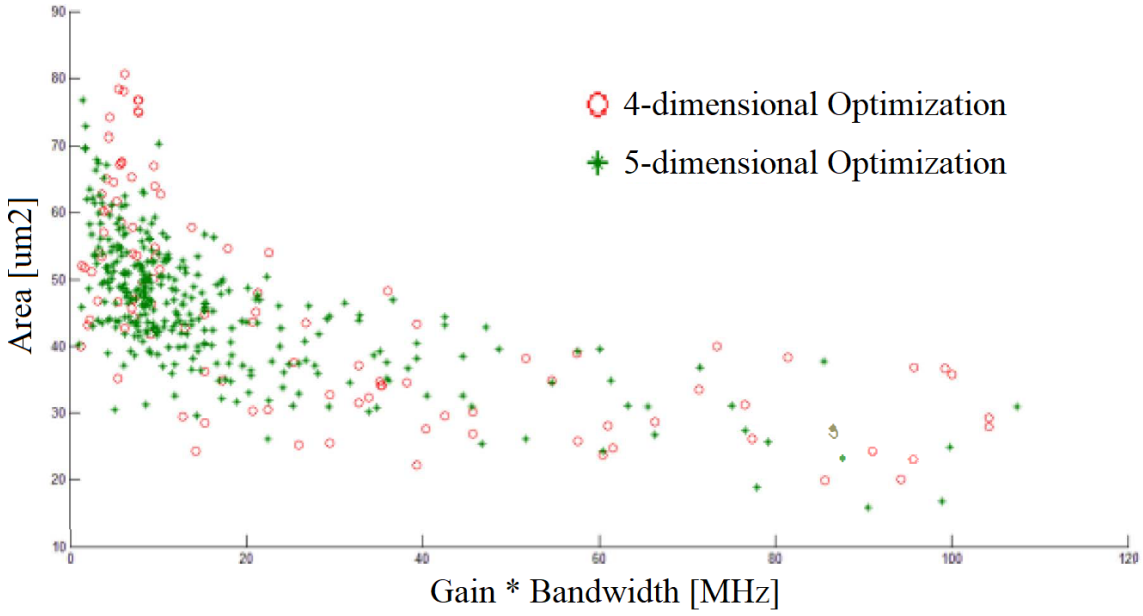

Figure 2.5. Two-dimensional Projections of the Four- and Five-dimensional Optimizations [29]

Table 2.4 shows the results for the comparisons of the Pareto Fronts obtained with three-dimensional and four-dimensional optimizations of the folded-cascode amplifier circuit. Table 2.5, on the other hand, shows the results obtained for the comparison of the four-dimensional and five-dimensional optimizations. IGD values are based on the approximated true Pareto Fronts. Spacing values in Schott's metric are calculated from the Pareto Front of each optimization itself. Coverage set values, finally, are obtained by directly comparing the two different Pareto Fronts in order to calculate the percentage of solution points dominated by the other front.

It can be seen from Table 2.4 that increasing the number of dimensions from three to four, does not have any impact on the dominance performance of the Pareto Fronts, as calculated by the Coverage Set metric. Moreover, the Schott's Spacing values are also very

close to each other, which means a very similar solution distribution performance has been obtained on the Pareto Fronts with different dimensions. As both Table 2.4 and Table 2.5 suggest, the IGD values are also comparable for the optimizations with different dimensions, with only a marginal performance deterioration.

Table 2.4. Comparison of three- and four-dimensional synthesis of the folded-cascode amplifier [29]

# of Objectives	Metric	Test 1	Test 2	Test 3	Mean
3	IGD	1.38	1.15	1.09	1.21
4	IGD	1.26	1.32	1.20	1.26
3	Coverage Set	27%	32%	22%	27%
4	Coverage Set	31%	28%	26%	28%
3	Schott's Spacing	0.35	0.34	0.34	0.34
4	Schott's Spacing	0.32	0.34	0.32	0.33

Table 2.5. Comparison of four- and five-dimensional synthesis of the folded-cascode amplifier [29]

# of Objectives	Metric	Test 1	Test 2	Test 3	Mean
4	IGD	1.32	1.28	1.28	1.29
5	IGD	1.42	1.38	1.36	1.38

Thus, the results show that, an increased number of dimensions for the optimization of the folded-cascode amplifier does not significantly deteriorate the quality of the Pareto Front; hence MOEA/D is powerful enough for the synthesis of the complex circuits and systems, such as yield-aware optimization of the analog circuits and MEMS.

3. VARIATION IN ANALOG AND MEMS COMPONENTS

3.1. Introduction to Variations and Their Impacts on Analog Circuits and MEMS

The largest threat to the continuation of the IC industry is the cost of designing new systems. The expense of creating the full set of masks for the production of a modern integrated circuit (IC) is usually in the range of millions of dollars, while the costs of the entire design process regularly reaches tens of millions of dollars. Shortcomings within a project can force the repetition of the manufacturing process, multiplying the mask generation costs. This cost caused by the re-spinning of a design puts pressure on engineers to fully verify their system before passing it on to production, which contradicts short time-to-market demands, generated by short product life cycles. Changes in technology have allowed major developments within the semiconductor industry, including the scaling of transistor sizes down to nanometers. It has been noticed, however, that the efficiency of the tools and methodologies used within the design industry is not keeping pace with this scaling, and that the complexity of new systems is increasing exponentially. The accuracy of these design tools and methods has also been affected by the scaling of transistor sizes. The number and placement of individual dopant atoms within the silicon produce differences in the performance of individual transistors, and these effects are increasing in significance. The random physical differences cannot be removed from the manufacturing process and so must be efficiently modelled and incorporated into the design process [34].

The same arguments are applicable not only to analog IC design, but also to MEMS design as well. For instance, the device geometry for mechanical sensors might be relatively large (especially compared to the transistor dimensions used in the scaled down IC manufacturing); however, even small variations on certain device parameters, such as the capacitive gap between the electrodes of an accelerometer, have a huge impact on the device performance. Consequently, considering these variations during the design process is a must in order to guarantee robust system designs for MEMS; hence, to avoid the re-spinning in the fabrication process.

3.1.1. Physical and Electrical Variations for Analog and MEMS Components

Process variations in IC and MEMS design, can roughly be classified into two different groups, which are the physical variations coming from the manufacturing steps or the electrical variations that these physical variations lead to.

The most important component for analog design is the transistor. As an example for the physical and electrical variations, let us consider the change of the implant dose (number of dopants) in the gate region which is a physical variation since it is caused by manufacturing nonuniformities. This physical variation results in a change at the threshold voltage of the device, which is an electrical variation. Besides the variations in the doping concentrations, deposition/etching nonuniformities (thickness variations) of the conducting (Al etc.) and isolating (SiO_2 etc.) layers are quite important for the device (transistor); hence, for the circuit performances. One of the most important sources of device variability is the lithography process. Variations in small feature sizes, like in the gate length, might result in high variations at the circuit performances by not only increasing or decreasing the capacitance and resistance values of the transistors, but also creating a mismatch between the different devices of the analog circuit.

Other typical components in electronic design are the passive devices such as capacitors, resistors, and inductors. The process variations mentioned above for the transistors are also valid for these passive components and these process variations cause high variability in the electrical device performance of these passive components as well. As an example, the MIM (Metal-insulator-metal) capacitor structure given in Figure 3.1. can be considered. The capacitance functionality is obtained by manufacturing two levels of metal layers which are separated by an insulator as a dielectric material. The actual capacitance value is an electrical parameter and is determined by the area where the metal layers overlap, as well as the thickness of the dielectric material. The physical variations on the patterning process, which are contributed both by lithography and etching variations, will create different overlapping areas of metal layers. Thus, the capacitance value, which is an electrical parameter, will vary accordingly as well. The same applies to the dielectric material thickness. The nonuniformities in the deposition process will cause thickness

variations during the dielectric deposition process; hence, the capacitance value will be affected.

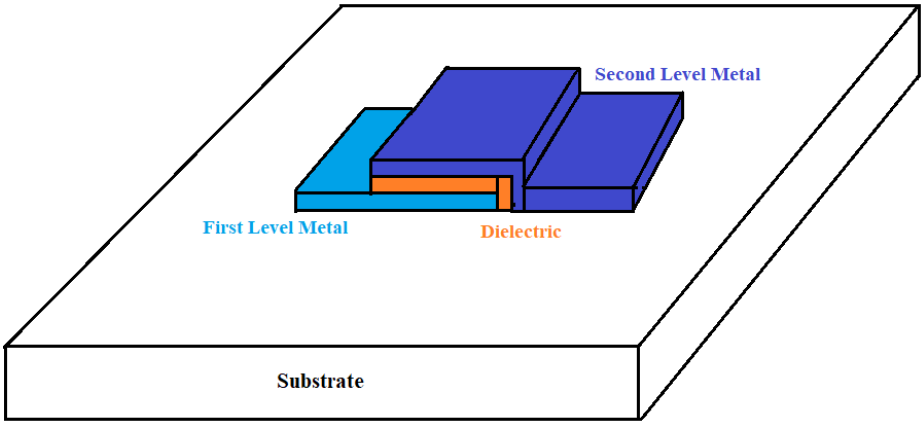

Figure 3.1. MIM Capacitor Structure

These passive components, together with the transistors, are the main devices used for analog IC design. Process variations in these individual devices will cause electrical variations for each single device, which will in turn be reflected on the circuit or system performance of the overall design.

MEMS design, on the other hand, is simply a heterogeneous design where a MEMS sensor or actuator is read-out or driven by an electronic circuit. Hence, the system consists of the mechanical and electrical signals. The variations on the electrical signal side have been mentioned above. The variations in the MEMS components are also not negligible and need to be considered for a robust system design that uses MEMS sensors or actuators. As an example, a capacitive MEMS accelerometer sensor given in Figure 3.2 can be considered. The sensor topology has two fixed (bottom and top) and one moving (the mass structure) electrodes. The mass structure is carried by the beam structure. Two different gaps at the top and bottom sides create a differential capacitance pair with respect to the acceleration direction; hence, an electrical signal can be generated from these capacitance changes. This is the working principle of the capacitive MEMS accelerometer sensor given in Figure 3.2.

Some of the design parameters of the MEMS sensor are the width, length, and thickness values for the beam and mass structures, as well as the capacitive gap between the electrodes. Formation of these structures include several process steps including deposition,

lithography, etching, and bonding. The physical variations coming from these process steps will directly alter the electrical performance of the sensor.

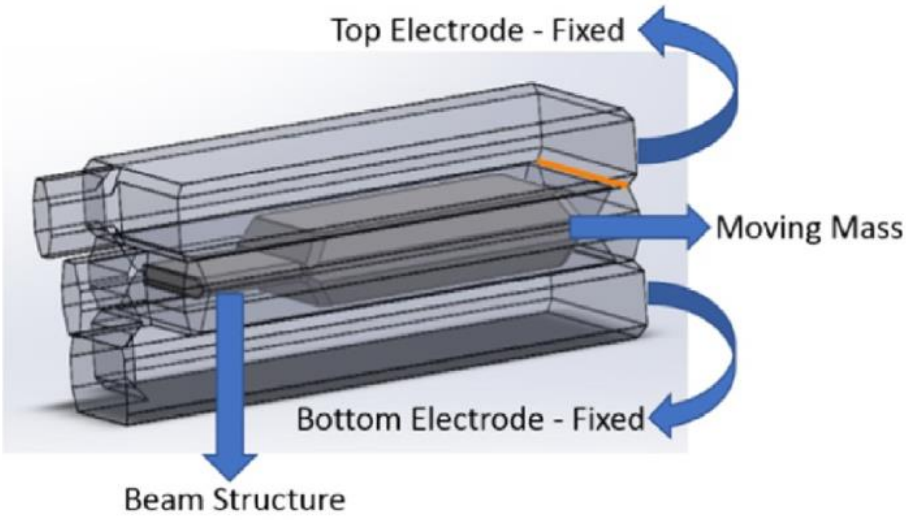

Figure 3.2. The MEMS Capacitive Accelerometer Sensor [6]

For instance, the thickness of the beam structure will directly impact how much the mass structure will move for a certain acceleration. The variations on the beam thickness will result in variations on the displacement of the mass for the very same acceleration; hence, the measurement accuracy of the sensor will be worsened.

3.1.2. Inter-die and Intra-die Variations

Process variations are either systematic or random. Systematic variations are deterministic in nature and affect each single structure equally. For example, the polysilicon gate width has a deterministic dependence on the spacing of neighboring polysilicon lines due to limitations of the lithography and the application of OPC methods [36]. Hence, the critical dimension (CD) difference between an isolated poly gate and a dense area poly gate will have a systematic error.

Random variations, on the other hand, are unpredictable in nature and include random variations in several parameters such as the device geometry, doping fluctuations, oxide thickness variations coming from the deposition process, etc. Analysis of the impact of deterministic variations on a certain circuit is relatively straightforward by using accurate

models [36]. Random variations, on the other hand, need to be statistically analyzed and these statistical definitions need to be considered in the design process.

Process variations can further be classified as inter-die and intra-die variations. Inter-die variations define the variations of the very same device in a single chip with the other die of the same wafer, from wafer to wafer and from lot to lot. On the other hand, intra-die variation refers to device characteristics that vary from device to device within the same die [37]. Often, intra-die variations exhibit spatial correlations, which means that the devices that are close to each other have a higher probability of being alike than devices that are placed far apart [36].

With increased process scaling, intra-die variation has become as important as inter-die variation when analyzing circuit or system performance and predicting the yield of a chip [37]. It means that the devices in the very same die cannot be treated as identical copies of each other.

3.2. Methodologies for Statistical Analysis in Analog ICs and MEMS

Process variations must be well-defined by using definitions such as the corner values and the standard deviations of the parameters in order to design a robust analog IC or MEMS. There are several techniques studied so far for statistical analysis of the integrated circuits and are also applicable to MEMS. The most common ones are the Monte-Carlo (MC) based methods and response-surface-based (RSB) methods. Both of these techniques have some advantages and disadvantages compared to each other. This trade-off generally occurs between the accuracy and the efficiency of the methodology.

3.2.1. Monte-Carlo-Based Techniques

Over the years, Monte Carlo simulation has become the standard technique for statistical simulation of circuits and for yield estimation during the design phase. At the nanoscale, no circuit parameter is truly deterministic; most quantities of practical interest present themselves as probability distributions. Thus, Monte Carlo techniques comprise the strategy of choice for statistical circuit analysis. The reason for this is that Monte Carlo

techniques are applicable to arbitrary circuits, arbitrary statistical models, and all performance metrics of interest, while allowing arbitrary accuracy. On the other hand, we gain the flexibility and accuracy of Monte Carlo simulation at the cost of speed: a single Monte Carlo run typically requires a high (up to thousands) number of sampling points (random) which results in a high computational cost [38].

There are several techniques to pre-define the Monte Carlo samples to reduce the overall computational cost. Instead of directly using random samples, these methods are used to create sampling points using a controlled random sequence such that the estimation accuracy can be improved. Latin hypercube sampling (LHS) is one of these techniques [39].

There is also an alternative approach which uses a completely different class of sampling methods called low-discrepancy sequences (LDS). Monte Carlo simulation that employs these deterministic sequences in place of pseudo-random sequences is termed as Quasi-Monte Carlo (QMC) [38, 40].

3.2.2. Response-Surface-Based Techniques

Response-surface-based techniques have been developed to overcome the efficiency problem of the Monte-Carlo with a sufficient accuracy. In practice, repeatedly running transistor-level simulations for so many times is often time-consuming and can even be infeasible for very large-size circuits. For example, running Monte-Carlo simulations for an industrial phase-locked loop (PLL) or analog-to-digital converter (ADC) will take months on a single machine. To overcome this issue, response surface modeling has been widely used to reduce the computational cost. The key idea with this method is to approximate the performance of interest (e.g., gain, power, etc.) as a polynomial function of the process parameters that are modelled as random variables (e.g., Vth, tox, etc.). Such a response surface model establishes an analytical dependence between device-level variations and circuit-level performance so that statistical analysis can be further applied to estimate the performance variation efficiently [41, 42]. There are also different approaches that use alternative techniques instead of the polynomial regression performed in standard response surface modeling. Kriging Metamodeling is one of these techniques [43].

3.3. Statistical Analysis of the Components used in Analog IC and MEMS Design

As mentioned in Section 3.1, circuit or system performance variations for analog IC and MEMS are caused by the electrical variations of each single component. These electrical variations, on the other hand, are caused by the physical variations coming from the nonuniformities of the manufacturing process.

The main components for analog IC design are the transistors (CMOS, Bipolar, etc.) as active devices and the passive devices such as resistors, capacitors, inductors, etc. For MEMS, on the other hand, the heterogeneous system is built by combining the mechanical sensor or actuator with a read-out circuitry, which is typically analog. Depending on the topology, some extra passive components (such as the feedback resistor and capacitance for a transimpedance based amplifier topology etc.) are also used. For MEMS, besides the analog IC components mentioned above, the electrical variations observed for the sensor (caused by the physical variations) also have a huge impact on the system performance and should be considered for robust system design.

Statistical analysis of several components has been performed or investigated to better understand the possible impacts at circuit or system level.

3.3.1. Statistical Analysis of the Components used in Analog IC

First of all, the process variations of the CMOS transistors and their impacts on the circuit performance have been investigated. For that purpose, the 0.35 μm process from AMS has been used [44]. The process models obtained from the manufacturer also include both inter-die and intra-die process variations to enable the statistical analysis at the circuit level. For the demonstration of the implemented methodologies, selecting one of these process variation libraries was sufficient, and intra-die variations have been reported in this study.

There are several process variations included in the model. The most important ones are the fluctuations on the gate oxide thickness, threshold voltage, and transistor width and length values.

In order to extract the impact of process variations of the 0.35 μm CMOS transistors on the circuit performance, 10,000 points Monte Carlo simulations have been performed for a certain design of the folded-cascode amplifier given in Figure 2.5. The simulation results for nominal parameters (with no variation) are given in Table 3.1. Table 3.2 shows the Monte Carlo simulation results obtained for intra-die variations.

Table 3.1. Simulation results of folded-cascode amplifier using nominal parameters

	Gain*Bandwidth (MHz)	**Gain (dB)**	**Phase Margin (°)**	**Power [mW]**
Values	16.78	79.14	84.45	0.538

Table 3.2. Simulation results of folded-cascode amplifier using intra-die variations

Intra-die Variations	**Gain*Bandwidth (MHz)**	**Gain (dB)**	**Phase Margin (°)**	**Power [mW]**
Min	15.81	72.04	84.44	0.5236
Max	17.30	82.44	89.87	0.5417
Mean	16.31	78.51	86.51	0.5334
Standard Deviation	0.21	1.45	1.66	0.004

After the process variation analysis of the CMOS transistors, the process variations of a SiGe Bipolar transistor have been studied [16]. In order to observe the electrical variations on the device given in Figure 3.3, the physical process variations from Yari-iletken Teknolojileri Arastirma Laboratuari (YITAL) cleanroom have been used.

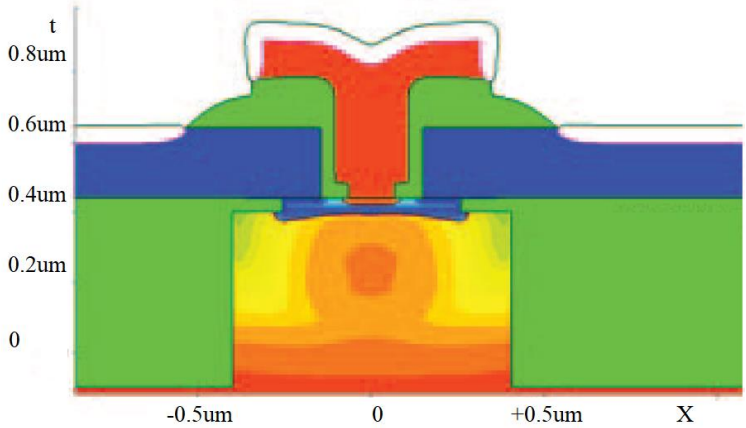

Figure 3.3. 0.25μm SiGe Bipolar Transistor

The device level simulations of the 0.25 µm SiGe Bipolar transistor have been performed using the TCAD simulator [45]. 100 different coefficient sets have been generated using QMC methodology to enable 100 different device simulations. Table 3.3 shows the physical variations (inter-die) obtained from the YITAL cleanroom for the bipolar transistor. Table 3.4 shows the electrical variations obtained from the 100 different TCAD simulations.

The results suggest that the 0.25 µm SiGe Bipolar transistor designed and simulated shows a robust performance for the cut-off frequency, DC operating point, and breakdown voltages; however, the variations in the current gain and the Early Voltage (*Va*) are considerable.

Table 3.3. Physical variations of 0.25 µm SiGe bipolar transistor (inter-die) [16]

Physical Variations – Process Step	**Inter-die variations [-3σ , +3σ]**
Burried Layer – As Implant	[-9% , +9%]
Extrinsic Base Poly – Boron Implant	[-7% , +7%]
Base Hole Lithography	[-10% , +10%]
Nitride Etching and Deposition for SIC (collector)	[-15% , +15%]
SIC Phosphorus Implant (collector)	[-8% , +8%]
SiGe Oxide Etch and Deposition	[-10% , +10%]
Base Carbon Implant	[-5% , +5%]
Base Boron Implant	[-5% , +5%]
Base – Emitter hole etching	[-8% , +20%]
Emitter As Implant	[-5% , +5%]
Emitter Poly Thickness	[-5% , +5%]
SiGe Layer Thickness	[-15% , +15%]

Table 3.4. Electrical variations of 0.25 μm SiGe bipolar transistor (inter-die) [16]

Parameter	Mean Value	3σ Value	Minimum	Maximum
Ft (GHz)	213.22	8.96	203.66	218.76
Beta maximum	4050	3840	2330	7760
***Va* (Volts)**	17.95	5.74	14.80	23.33
***Ic* (mA)**	0.89	0.48	0.64	1.40
***Vb* at max freq (Volts)**	0.93	0.09	0.89	1.04
CE Breakdown (Volts)	2.32	0.29	2.16	2.52
Beta at operating point	337.33	185.21	225.32	490.49

A similar process variation analysis has also been performed for a 1 nH spiral inductor. This passive device has been designed at YITAL cleanroom and simulated using the Sonnet SW [46]. 20 different simulations have been realized by altering the physical parameters using QMC for the values given in Table 3.5. The variations at electrical domain, as the outcome of the simulations, are given in Table 3.6.

Table 3.5. Physical parameters and variations of the 1 nH spiral inductor

Physical Parameter	Mean Value (μm)	3σ Value (μm)
Dielectric thickness	5	0.24
Top metal thickness	1.5	0.15
Top metal line width	10	0.15

Table 3.6. Electrical variation analysis of the 1 nH spiral inductor

Parameter	Minimum	Maximum	Mean	3σ Value
L value (nH)	0.986	1.031	1.008	0.54
Q max	14.81	16.46	15.33	1.62

In order to include the impact of the variability of a passive device on the circuit performance (not only the transistor variability impact as performed for the folded-cascode amplifier example), a low-noise amplifier (LNA) circuit used for 900 MHz GSM applications has been simulated. The amplifier topology used (designed at YITAL) is not given due to confidentiality constraints.

For the statistical analysis of the LNA, the ideal 1 nH inductor in the design has been replaced with the s-parameters of the 20 different inductors simulated and described above. Each of these 20 simulation results of the inductor have been used 5 times with different variability sets of the 0.35 μm CMOS technology; hence, in total 100 different variability simulations have been run.

The results show the S21 gain (peak value) and the Noise Figure (NF) for 100 different simulations.

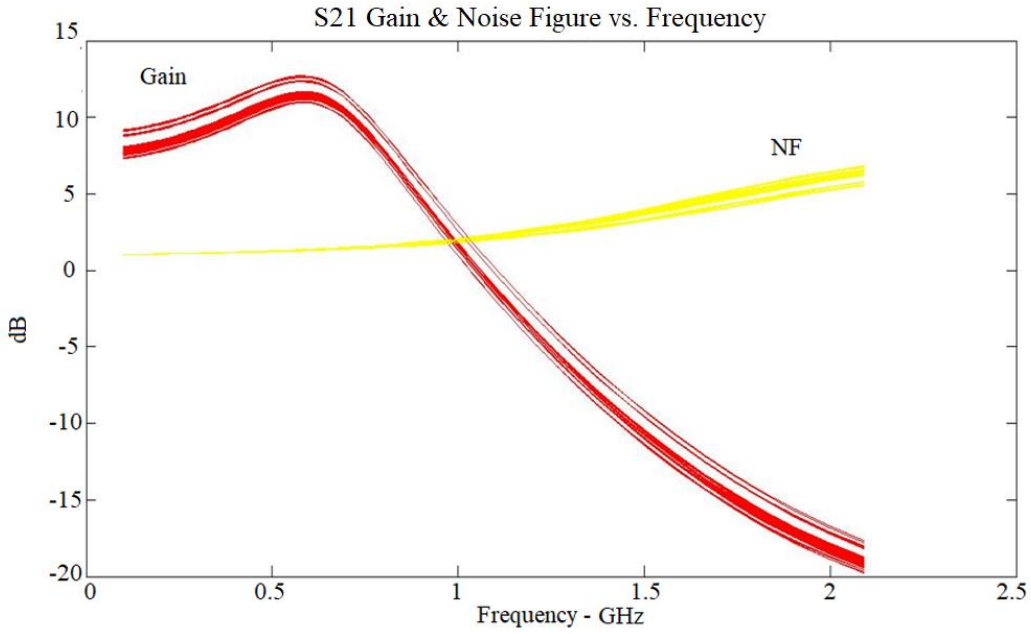

Figure 3.4. S21 and NF Simulation Results of the LNA for 100 Variability Simulations

Table 3.7 Statistical analysis of the LNA at a certain frequency

[dB]	Minimum	Maximum	Mean	3σ Value
Peak S21	10.93	12.71	11.58	1.47
Noise Figure	1.34	1.26	1.31	0.06

Several simulations for analog IC have been run to demonstrate the impact of the process variations on the circuit performance. The folded-cascode amplifier circuitry will be used as one of the main test circuits in this thesis. Variability simulations for the LNA, on the other hand, have been run to introduce the variability in RF circuits in case a future work on yield-aware optimization of RF circuits is carried out.

3.3.2. Statistical Analysis of the Components used in MEMS

MEMS design is a heterogeneous design where a mechanical sensor or actuator is driven or read by an electronic circuit, which is mostly analog. Statistical analysis of the components in analog IC and their impacts on circuit performance have been investigated above. The major mechanical component for the MEMS is the sensor or the actuator itself. In this part, the statistical analysis of a capacitive MEMS accelerometer sensor, given in Figure 3.2, has been performed.

The design variables of the sensor are given in Table 3.8. The physical variations of these design variables and the sources of the variations in terms of the manufacturing step, are also given. *L* refers to the length, *W* to the width, and *t* to the thickness of the mass and beam structures. C_{gap}, on the other hand, is the capacitive gap between the fixed and the moving electrodes.

Table 3.8. Design variables and physical variations for the MEMS sensor

Design Variables	**Process Steps**	**Physical Variations [-3σ , +3σ]**
L_{mass}	Lithography + Etch	[-1.4% , +1.4%]
W_{mass}	Lithography + Etch	[-1.4% , +1.4%]
t_{mass}	Si Etch	[-1.5% , +1.5%]
L_{beam}	Lithography + Etch	[-1.4% , +1.4%]
W_{beam}	Lithography + Etch	[-1.4% , +1.4%]
t_{beam}	Si Etch	[-1.5% , +1.5%]
C_{gap}	Sacrificial SiO2 deposition	[-3% , +3%]

It should be noted that the physical variations mentioned for the mass and the beam structures are related to the etch variations of the etched Si layer. The actual thickness variation of the beam and mass are calculated based on the remaining material. For instance, considering that the 400µm thickness of Si for a 500µm wafer is etched to obtain the 100µm thick beam structure. In that case, the etch material thickness might vary (for 3σ) from 394µm to 406µm, which means the remaining beam thickness will vary in between 94µm and 106µm thickness values; hence, the 3σ variation will be equal to 6%. In case where the etched material is thinner than the remaining material, than the 3σ variation at the beam thickness will be lower than 1.5%.

70 different QMC samples have been used to evaluate the electrical variability of the MEMS sensor using the physical variations given in Table 3. The evaluation of the sensor performance is done by using an accurate analytical model given in Chapter 6. The working principles and the fabrication details of the sensor are also given in Chapter 6.

Figure 3.5 shows the electrical variations for the noise and measurement range performances for the accelerometer sensor. The sensor noise is equal to 19.08 µg/$\sqrt{Hz}$ and the measurement range of the sensor is equal to 97.8 g for the nominal parameters.

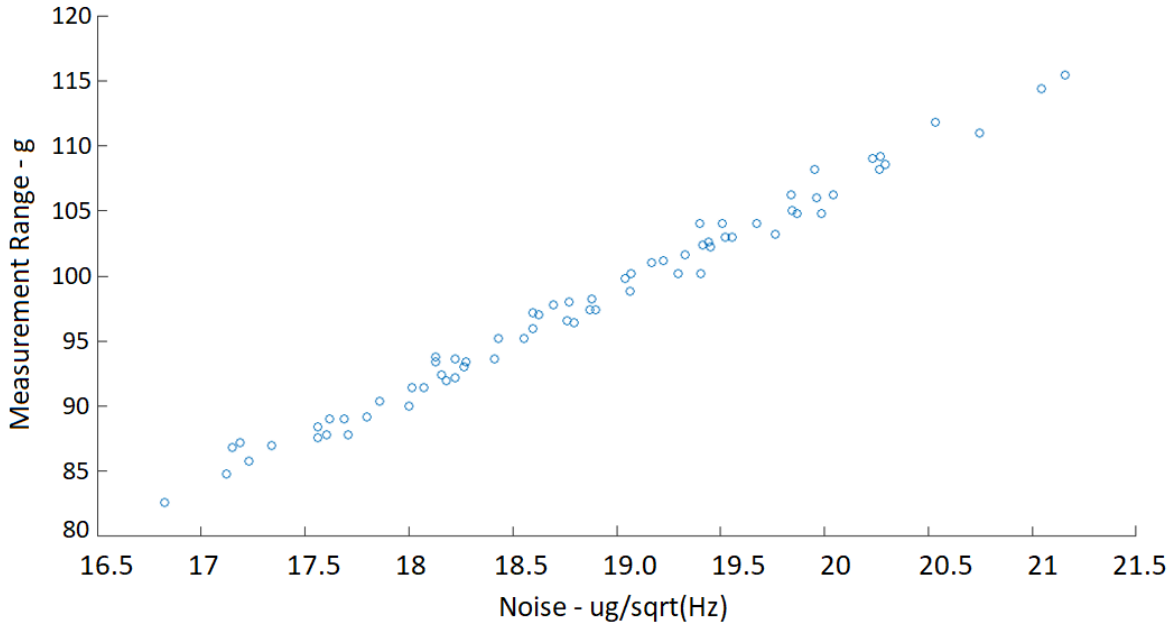

Figure 3.5. The Electrical Variations for the Noise and Measurement Range

The noise for the 70 QMC simulations varies between 16.71 $\mu g/\sqrt{Hz}$ and 21.23 $\mu g/\sqrt{Hz}$; while the measurement range varies between 82.4 g and 115.6 g values.

The variability models of the other passive components, such as resistor and capacitor, have also been implemented in the yield-aware optimizations performed and will be given in more detail in Chapter 6. The assumption for the fabrication of the resistor was a poly resistor where the physical variations such as the poly thickness, etc., are causing the change of the resistance value. For the fabrication of the capacitor, a MIM capacitor topology has been considered. The variations in the capacitance value are caused by the physical variations such as the deposition nonuniformities of the dielectric material, etc.

As a result, the statistical analysis of different components used in analog IC and MEMS design have been performed by simulating the variabilities in the process to obtain the electrical variations of these active and passive devices. Moreover, this electrical variability information of the components has been used to perform circuit level simulations in order to observe the impact on the circuit performance.

4. IMPLEMENTATION OF DIFFERENT YIELD OPTIMIZATION METHODOLOGIES

4.1. Review of Different Yield Optimization Methodologies

Yield-aware optimization requires the implementation of two main steps. The first one is the technique to be used for yield estimation and the second one is the technique to integrate the estimated yield value into the optimization algorithm. In Chapter 3, several methodologies for statistical analysis have been discussed. Statistical analyses are performed in order to identify the performance variability of the optimized circuit or system. This variability information, together with the pre-defined acceptance regions, define the yield value of a certain design point. Figure 4.1 illustrates this yield calculation. Let us consider a design point at value 10 for the Performance 1 and 100 for Performance 2 (for the nominal process parameters) and assume that both performances are objectives to be maximized, such as circuit gain, phase margin, etc. The nominal design point is shown in red in the figure. The performance variations, on the other hand, are calculated by performing 100 different performance evaluations using the process variations; thus, creating the performance variation cloud given in Figure 4.1 in green.

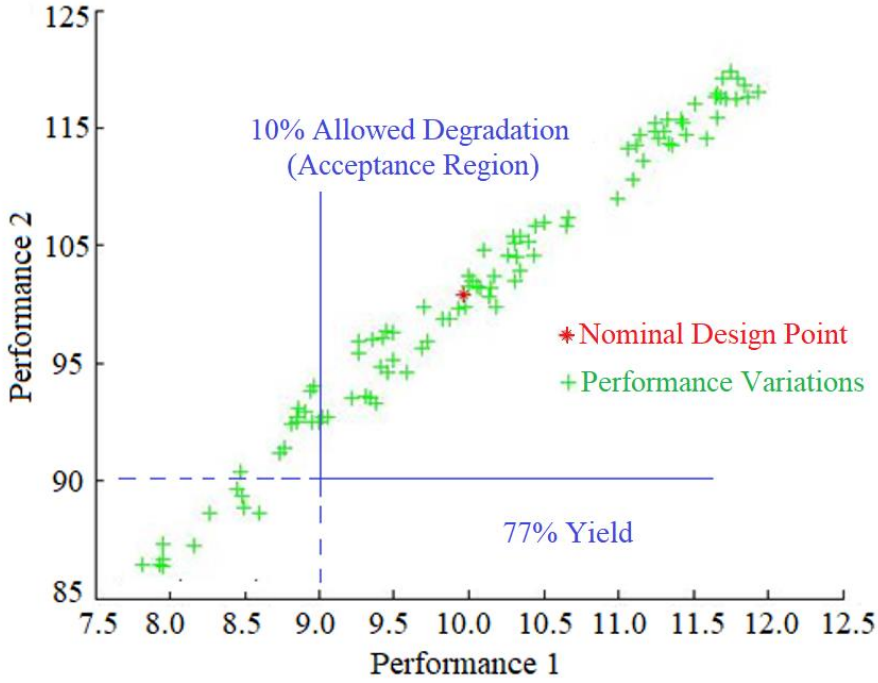

Figure 4.1. Illustration of the Yield Calculation of a Certain Design Point

Defining the acceptance region as 10%, any solution with a Performance 1 value below 9 or Performance 2 value below 90, will be considered as a design point which is not acceptable. In Figure 4.1, 77 out of 100 solutions are within the acceptance region, meaning that the yield value for this design is 77%.

Multi-objective yield-aware optimization tools require a high number of performance evaluations (different iterations, statistical analysis of each solution etc.) which slows down the process in case the evaluations are performed by simulations rather than by using other methodologies such as response surfaces. At the nanoscale, no circuit parameters are truly deterministic, and most quantities of practical interest present themselves as probability distributions. Thus, simulation techniques based on Monte Carlo comprise the strategy of choice for statistical circuit analysis. The reason is that Monte Carlo techniques are applicable to arbitrary circuits, arbitrary statistical models, and all performance metrics of interest, while allowing arbitrary accuracy. This flexibility and accuracy for yield estimation is obtained at the cost of a high number of Monte Carlo simulations.

There have been several studies to speed up the Monte Carlo approach by reducing the sample size to be used for statistical analysis while keeping the high yield estimation accuracy. In the typical use, a pseudorandom sample generator is employed to generate the Monte Carlo sample sequence. However, there is a completely different class of sampling methods called low-discrepancy sequences (LDS). In contrast to standard pseudorandom sequences, these are deterministic sequences with no random component. The points in the sequence are generated to satisfy some rigorous notion of uniform coverage of the sampling space. The Monte Carlo simulation that employs these deterministic sequences in place of pseudo-random sequences is, as mentioned in Chapter 3, known as QMC.

The yield estimation methodology of the implemented yield-aware optimization algorithm is based on performance evaluations obtained via the QMC sample sets. The selection of the QMC sample size determines the accuracy of the yield estimation. Variation space is the solution space obtained by running the variability simulations. Very low sample sizes will not be able to capture a sufficient performance , causing an inaccurate yield estimation. A very high sample size, on the other hand, will increase the number of simulations; hence, the overall optimization time will not be practical anymore. An ideal way of selecting the QMC sample size is to decrease the sample size as much as possible,

but also guarantee that the sampling leads to a good approximation of the probability distribution function. For that purpose, several QMC sample sizes for different optimization problems have been compared by checking the variability information reflected on the different objectives. These data are given in Chapter 5 and Chapter 6.

Appropriately evaluating the performances and sampling the variation parameter space for accurate yield estimation is only one aspect of the problem. A second, and even more important, aspect for implementing a yield-aware optimization tool is embedding these performance variations, hence the yield value, within the design process.

Several techniques have been developed for consideration of robustness into the optimizer [47]. These different methodologies can be classified in yield-aware synthesis techniques [48-49], and yield-optimized synthesis techniques [50].

Yield-optimized synthesis is commonly known as design centering and tries to make the design tolerant to variability by maximizing the distance to all constraint boundaries. On the other hand, yield-aware methodologies are based on using the yield information of the candidate solutions directly as a constraint or an objective, in order to increase the robustness of the designs.

4.2. Implementation of the Multi-Objective Yield-Aware Optimization Tool

Before discussing the implementation of the methodology, the decision between yield-aware and yield-optimized techniques need to be made. The purpose of the implemented tool is optimizing the yield for several optimal design points; hence, the yield of a PF. Yield-aware optimization methodologies are suitable to be implemented for multi-objective optimization. Yield-optimized techniques, on the other hand, are mostly focused on the design centering of a pre-defined performance specification set, which is simply an optimization of a single design point, rather than a PF where the trade-offs between different performances are studied. Thus, yield-aware optimization methodologies have been selected for the multi-objective yield optimization tool developed in this work.

Another question to be answered before the implementation of the tool is when to introduce the yield-aware optimization in the design process. The answer has been

empirically studied in the following chapters. Starting the yield-aware optimization from the very beginning slows down the synthesis process drastically as the number of simulations for yield estimations is high. An alternative consists in running a nominal optimization for a certain number of iterations and then use the obtained design solutions as starting points for the yield-aware optimization. These two phases are called nominal optimization phase and yield-aware optimization phase.

Running a nominal optimization first with a low number of iterations, robust solutions can be easily generated by the algorithm; however, the quality of the solutions are very low. On the other hand, running too many iterations in the nominal optimization phase, decreases the probability of replacing the solutions in the PF with a robust solution by so-called "freezing" the design points. Hence, the number of iterations for the nominal optimization phase needs to be selected carefully. A detailed study has been performed to define when to switch from nominal to yield-aware optimization. Details of this work are given in Chapter 6.

In the thesis, yield-aware multi-objective optimization of a differential folded-cascode amplifier and a capacitive MEMS accelerometer system has been performed. For both test cases, a generic multi-objective yield-aware optimization methodology has been used. The main differences to set up the robust optimization flow for the analog circuit and the MEMS system are the simulation files or models used for performance evaluations, design parameters, and selection of the performances to be optimized.

In this generic synthesizer, the optimization includes two different phases. The first phase is the nominal optimization of the circuit or system to be optimized while the second one is the yield-aware optimization where the robustness of the pre-optimized solutions is improved. As shown in Figure 4.2, the algorithm runs for several iterations without including the yield concept at the beginning of the optimization, in order to create nominally optimized solutions before enhancing the yield. Once the nominal optimization is halted, all objectives are multiplied by a factor which we call the "transition coefficient". For the optimization, MOEA/D [11] is used as mentioned in Chapter 2. The MOEA/D algorithm replaces existing solutions by newly generated ones only in case there is an improvement in the fitness function for that particular subproblem. By using the novel "transition coefficient" technique (multiplication of each objective value with a certain constant at the end of nominal

optimization) in our methodology, the replacement of higher performance solutions by those which have slightly worse performance than the existing ones but with improved robustness, is promoted. After the update of the objective values using the transition coefficients (details are given in Chapter 5 and Chapter 6), the yield-aware optimization phase is run for several iterations to generate yield-aware PFs [4].

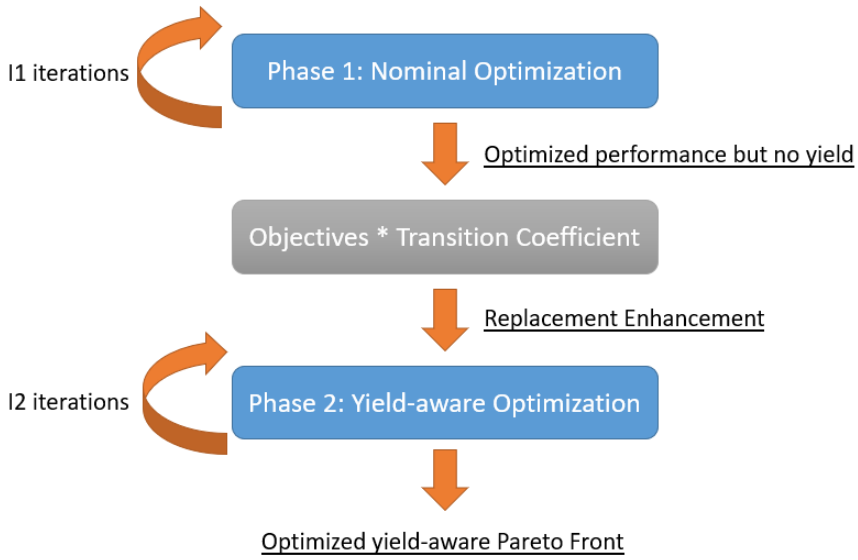

Figure 4.2. Two-step Yield-aware Optimization Methodology Implemented [4]

The selection of the design parameters, process variations to be included, circuit or system performances to be optimized are problem-specific inputs for the optimizer. However, the optimization parameters such as the stopping criterion for the nominal optimization, the selection of the transition coefficient, and the QMC sample size should be as generic as possible to avoid extra burden for the user. For this purpose, the multi-objective yield-aware optimization tool has been progressively improved to develop a more generic tool.

Initially, for the two-dimensional optimization of the folded-cascode amplifier, all of these parameters were fixed values based on several empirical tests and also the specific design knowledge. For three-dimensional yield-aware optimization of the folded-cascode amplifier, on the other hand, a technique based on a linear increment of the QMC sample size has been applied at the yield-aware optimization iterations. The main idea was increasing the number of samples, because of the increased number of objectives, as the variation space of three dimensions has been considered as more difficult to capture than the variation space of two dimensions. Using the sample size method defined above, the QMC

sample size per solution starts with an acceptable[1] but small number and goes up to a higher number at the last iteration, with a linear increment of the sample size per generation. With this technique, replacements by robust solutions are done at the initial iterations while the final iterations with higher QMC sample size guarantee the accuracy of the yield estimation.

For the stopping criterion of the nominal and yield-aware optimizations, on the other hand, a technique based on checking the improvement of the PF has been implemented. For that purpose, a generic methodology based on monitoring the improvements of the nominal PF to avoid the pre-mature transition to yield-aware optimization phase and the solution freezing risk, has been developed and used in the yield-aware optimization of the MEMS accelerometer. A similar technique has been applied during yield-aware optimization phase to serve as a stopping criterion for the overall optimization.

Once the stopping criterion is met and nominal optimization is finished, all objectives are multiplied by the transition coefficient in order to increase the replacement probability of the robust solutions during the yield-aware optimization. If the transition coefficient number is set too close to 1, which means keeping the original objective values, the probability of replacement will be too low. On the other hand, if the transition coefficient is much smaller than 1, the probability of replacement will dramatically increase; however, the quality of the solutions will not be good. Hence, a very high number of iterations at the yield-aware optimization phase will be needed to obtain an optimal PF. Some empirical studies have been carried out to determine the best transition coefficient number and reported in Chapter 6. The summary of the improvements carried out to make the tool more generic is given in Table 4.1.

Table 4.1. The progressive development of the optimization parameter selection

Yield-Aware Optimization Performed	**QMC Sample Size**	**Number of iterations for nominal optimization**	**Transition Coefficient**	**Number of iterations for yield-aware optimization**
Two-dimensional: Folded-Cascode Amplifier	Fixed	Fixed	Fixed	Fixed
Three-dimensional: Folded-Cascode Amplifier	Incremental	Fixed	Fixed	Fixed
Capacitive MEMS Accelerometer System	Incremental	PF Improvement Monitoring	Empirically Determined	PF Improvement and Diversity Monitoring

[1] Details of what acceptable is given in Chapters 5 and 6

The studies related to QMC sample size selection and its impact on the accuracy of the yield estimation, the details of the PF improvement based stopping criterion for the nominal optimization phase, and the tests performed to select the optimal transition coefficient and also to address the trade-off between the solution quality and the convergence speed to the robust PF are discussed in Chapters 5 and 6 rather than this chapter in order to support the methodologies with actual data.

The goal of multi-objective yield-aware optimization tools is to synthesize robust PFs where the yield of each design point on the front is optimized. In order to enhance the yield, performance variations of the design points need to be reduced. To achieve this, three different yield-aware optimization methodologies have been implemented and compared. These methodologies differ from each other by the constraint mechanisms used to optimize the yield. The illustration of these three different techniques using the variation space of a single solution is given in Figure 4.3. The single circle in the figure represents the performances of one solution for nominal technology process parameters; while the crosses on the PF represent the performances of the same solution for a set of samples of the technological process variations.

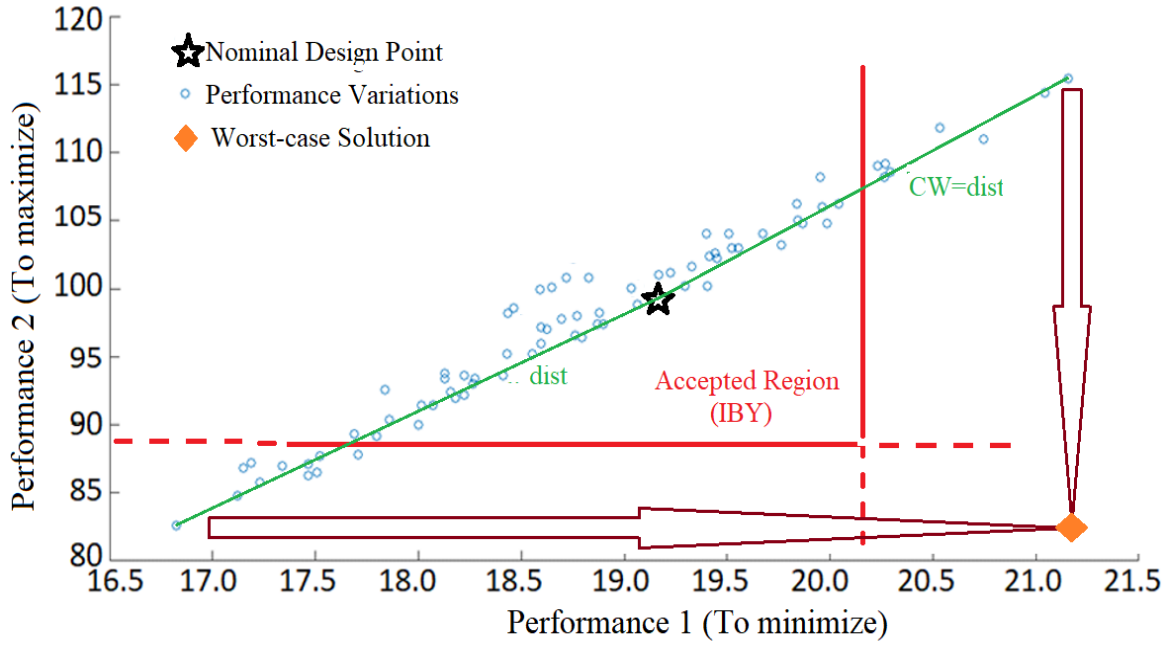

Figure 4.3. Demonstration of the Three Yield-aware Optimization Techniques

4.3. Techniques for the Integration of the Yield-Awareness into the Optimizer

The details of these three methodologies are as follows.

4.3.1. The CW Technique

The cloud width (CW) technique is a novel technique proposed in this thesis and is based on the calculation of the Euclidean distances among different variation samples of a certain design point. The maximum distance between two different sample points is defined as the CW of that particular design point as illustrated in Figure 4.3.

If N is the number of variation samples for each solution point, j is the index for the related sample point, i is the index for the design point on the PF, m is the number of objectives and finally l is the index for the objective, the CW of each single solution point i is calculated as:

$$CW_i = \max_{1 \le j \le N} \left\{ \max_{(j+1) \le k \le N} \left\{ \sqrt{\sum_{l=1}^{m} \left(x_{i,j,l} - x_{i,k,l} \right)^2} \right\} \right\} \tag{4.1}$$

To avoid the dominance of any objective value in the Euclidean distance calculation, CW is calculated by normalizing performance values of the variation samples with respect to the nominal solution points.

4.3.2. The IBY Technique

The IBY technique is based on defining an acceptance region for the nominal solutions by allowing a certain amount of performance degradation. This is illustrated in Figure 4.3 by the acceptance region boundary lines plotted in purple. The yield value for each design point on the PF is calculated by checking the percentage of the variation samples covered by this acceptance region and the solution is penalized by adding an extremely high value if the pre-defined yield constraint is not satisfied.

If p is the population size, i is the index for the design point on the PF, m is the number of objectives, k is the index for the objective, d_k is the user-input for variation percentage allowed for the k^{th} objective, N is the number of variation samples for each solution point, and finally j is the index for the related sample point, the yield value for each design point on the PF is calculated as:

For all $1 \leq i \leq p$ of each $1 \leq j \leq N$,

$$Y_{i,j} = 1 \quad \text{if} \begin{cases} x_{k,i,j} \geq (1 - d_k).x_{k,i}^{nom}, for\ all\ maximization\ objectives \\ and \\ x_{k,i,j} \leq (1 + d_k).x_{k,i}^{nom}, for\ all\ minimization\ objectives \end{cases} \tag{4.2}$$

$$Y_{i,j} = 0 \ , otherwise \tag{4.3}$$

And the yield for each solution is calculated as:

$$Y_i = \frac{\sum_{j=1}^{N} Y_{i,j}}{N} \tag{4.4}$$

4.3.3. The WCPF Technique

The last technique is called worst-case Pareto-front (WCPF) and is based on choosing the worst objective values among the variation set (as illustrated in Figure 4.3) of each single solution in order to generate a worst-case solution [51]. With this technique, rather than defining a yield related constraint, the optimization is performed using the PFs formed by the worst-case scenario of each solution (meaning worst case objective values among all samples), after each iteration. The entire PF obtained with worst case solutions is called worst-case pareto front (WCPF).

These three techniques have been implemented and compared in Chapters 5 and Chapter 6.

5. YIELD-AWARE OPTIMIZATION A FOLDED-CASCODE AMPLIFIER

The first circuit synthesized using the yield-aware optimization methodologies given in Chapter 4 is the folded-cascode amplifier. For yield evaluation, the process variation library of 0.35µm CMOS technology (AMS) has been used. The process variations used are given by the fluctuations on the thickness of the gate oxide, threshold voltage, and the transistor geometry for the width and length values. For the 13 transistors in the circuit, 52 different process variations are, therefore, included in the synthesis loop to be used to simulate the performance variability of the circuit for different design points.

The three techniques in Chapter 4 (CW, IBY, and WCPF) are all implemented for the robust optimization of the folded-cascode amplifier and compared for both two- and three-dimensional optimization.

5.1. Design Variables and Objectives

In the amplifier circuit, there are in total 13 transistors, whose parameters (e.g., the gate length of a transistor) are the design variables used for the robust circuit search. As also mentioned in Chapter 2, because of the design characteristics (i.e., symmetrical geometry of the current mirrors), some of the transistor dimensions are forced to be equal to each other. There are 11 design variables used for yield-aware optimization of the folded-cascode amplifier and they are identical to Table 2.1. Allowable ranges of these design variables, on the other hand, are different compared to the ones given in Table 2.2, as the technology used is 0.35µm CMOS instead of the previously used 0.18µm CMOS due to the availability of the process variation information of the technology models. The allowable ranges for the design variables are given in Table 5.1.

The objectives and constraints applied for the two- and three-dimensional yield-aware optimization are given in Table 5.2. For three-dimensional optimization, two different sets of objectives have been tested. The transistors are forced to operate in the saturation region and this is imposed as a constraint.

Table 5.1. Allowable ranges of the design variables of the folded-cascode amplifier

Type of the design variable	Minimum Value	Maximum Value
W	500nm	800µm
L	350nm	10µm
Ib	0.5µA	2.5mA

Table 5.2. Objectives and constraints for yield-aware optimization of the folded-cascode amplifier

Performance	2D Robust Synthesis	3D Robust Synthesis Test 1	3D Robust Synthesis Test 2
Gain-Bandwidth Product	Objective #1	Objective #1	Objective #1
Gain	Objective #2	Objective #2	Objective #2
Phase Margin	Constraint	Objective #3	Constraint
Power	Constraint	Constraint	Objective #3
Area	Constraint	Constraint	Constraint
Operating Points	Constraint	Constraint	Constraint

5.2. Determination of the Optimization Parameters

For nominal optimization, the determination of some basic optimization parameters (e.g., the population size, number of iterations, etc.) is needed. For the yield-aware optimization, on the other hand, there are additional optimization parameters.

At the beginning of the optimization (during the nominal optimization phase), the algorithm runs for several iterations without including the yield concept, in order to create nominally optimized solutions before enhancing the yield. Hence, the first parameter to determine for the yield-aware optimization is the number of iterations used for the nominal optimization phase. Starting yield-aware optimizations from the very beginning slows down the synthesis process drastically due to the increased number of simulations performed for yield estimations. Using fewer iterations than optimal, allows the algorithm to generate robust solutions easily, but the quality (nominal performance values) of these solutions are very low. On the other hand, setting this number too high decreases the probability of replacing the solutions in the PF with a robust solution by so-called “freezing” the design points already in the first phase. For the yield-aware optimization of the folded-cascode

amplifier, some quick empirical checks have been run to define the number of iterations [17]. For two-dimensional optimization, 75 iterations have been run for the nominal optimization phase, while this number is equal to 180 for the three-dimensional optimization. For the two-dimensional optimization, 15 iterations for the time-consuming yield-aware optimization phase was sufficient in order to obtain a PF with robust, well distributed and diverse solutions. For three-dimensional optimization, on the other hand, 20 iterations at the yield-aware optimization phase have been performed to guarantee a PF with a reasonable number of robust solutions. For the selection of these fixed iteration numbers, Section 4 of this chapter and [17] can be referred.

As explained in detail in Chapter 4, after performing the nominal optimization, all objectives are multiplied by a factor called transition coefficient. For the robust optimization of the folded-cascode amplifier, the constant 0.9 has been set for both two- and three-dimensional yield-aware optimizations.

Once the update of the objective values using the transition coefficients are performed, one of the three different yield-aware optimization methodologies mentioned in Chapter 4 (CW, IBY and WCPF) are run for several iterations to generate yield-aware PFs [17].

The CW methodology, as shown in Figure 4.3, is based on limiting the maximum Euclidean distances in the performance variation space of a certain design point. During the checks for comparing different QMC sample sizes, it has been observed that the average CW values are 0.3 (a normalized span of 30%) for two-dimensional solutions and 0.4 (a normalized span of 40%) for three-dimensional solutions. A certain CW value has been set as a constraint for the integration of the robustness into the optimizer. This value have been defined as half of the CW values mentioned above (which are obtained after nominal optimization); hence, 0.15 for two-dimensional optimization and 0.2 for three-dimensional optimization, to guarantee that the variability of the generated solutions are improved by 50%.

For the robust optimizations performed using the IBY methodology, the acceptance region to calculate the yield is defined as 5% performance degradation compared to the

nominal performance value of each objective. A practical value of 80% has been used as the constraint for IBY to decide whether a penalty will be applied to the solutions on the PF. This value has been selected relatively low in order to increase the probability of replacing a solution which is not robust (meaning a solution with less than 80% yield).

Since the last technique WCPF is based on choosing the worst solution among the variation set of each single solution, no specific optimization parameters are required.

5.3. QMC Sample Size Selection for the Folded-Cascode Amplifier

The selection of the sample size for QMC for the yield-aware optimization of the folded-cascode amplifier is based on several experimental results. The first experiment checks the variation percentages of each objective with different sample sizes, in order to observe the coverage of the variability space for different design points. For that purpose, three different design points obtained by two-dimensional nominal optimization of the circuit are compared by running variability simulations using different QMC sample sizes, as well as a 10,000-sample standard MC simulations to be used as an accurate reference for yield estimation. The closer the variation percentages exhibited by any methodology to the MC simulation, the more accurate the yield estimation is [17]. The comparison data, which shows the variability percentages of each objective for each solution point, is given in Table 5.3.

Table 5.3. Experimental data for the selection of QMC sample size for folded-cascode amplifier

Sampling Method	**Solution # 1: %Var. of Obj1/Obj2**	**Solution # 2: %Var. of Obj1/Obj2**	**Solution # 3: %Var. of Obj1/Obj2**
MC – 10,000	13.20 / 1.61	34.22 / 2.06	4.53 / 1.91
QMC – 500	12.88 / 1.57	33.63 / 2.02	4.42 / 1.86
QMC – 100	12.66 / 1.53	32.91 / 1.98	4.33 / 1.81
QMC – 75	12.53 / 1.49	32.35 / 1.97	4.31 / 1.80
QMC – 50	12.43 / 1.48	31.85 / 1.97	4.28 / 1.79
QMC – 25	11.85 / 1.36	29.75 / 1.92	4.12 / 1.74

The table shows that, in terms of the objective variabilities captured, there is a big difference between 25 and 50 QMC samples. For the sample sizes above 50, on the other hand, the improvement is relatively small. Hence, it can be concluded from the table that 50 QMC samples is good enough for a coarse approach for yield estimation and this number should be increased to the range of hundreds for highly accurate results [17].

The analysis of variation percentages already gives an idea about the quality of the sampling; however, it is not sufficient to guarantee that the QMC sampling leads to a good approximation of the probability distribution function.

For that purpose, the distribution of the MC results of the circuit performance values have been analysed and several QMC sample sizes have been compared in that manner. In order to enable this check, a probability plot technique called quantile-quantile plot, also known as QQ-Plot has been used [52]. The QQ-Plot is a graphical method for comparing two probability distributions by plotting their quantiles against each other.

Figure 5.1 shows the 10,000-sample MC histogram of the Gain-Bandwidth Product of a solution, which looks like a good approximation to a Gaussian distribution. QQ-Plot for 10,000-sample MC simulation versus the normal distribution function has been plotted in Figure 5.2, also showing the Gaussian distribution behaviour, as the MC samples can capture the red line showing the quantiles of an ideal normal distribution function [17].

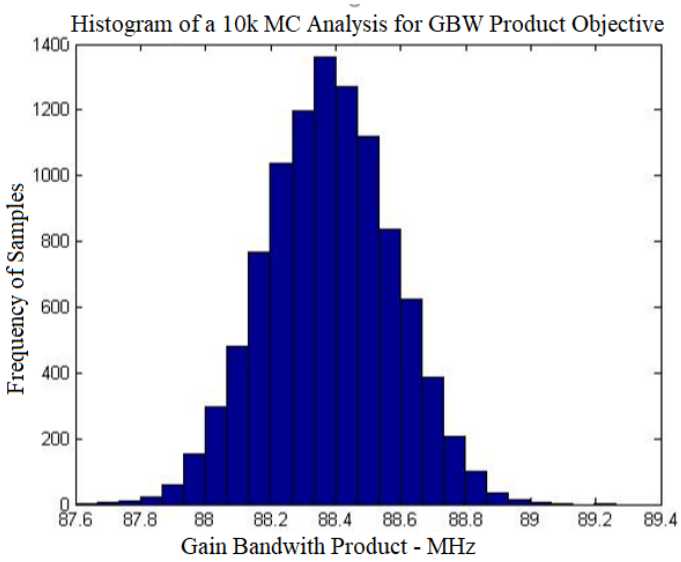

Figure 5.1. Histogram of a 10000-sample MC Analysis for GBW

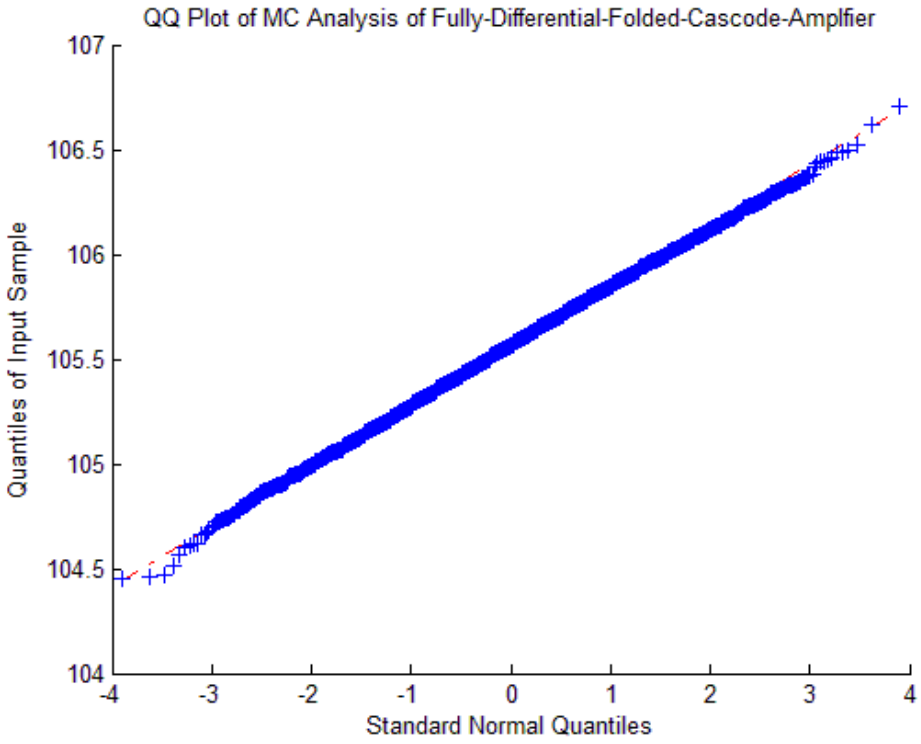

Figure 5.2. QQ-Plot of a 10000-sample MC Analysis for GBW

Besides comparing a 10,000-sample MC simulation with the ideal Gaussian distribution, different QMC sample sizes have been used for distribution comparisons in order to see whether the distribution behaviour is kept.

In Figure 5.3, the QQ-Plots are directly compared using two different data sets with a common distribution. The reference data set, in each comparison, is the 10,000-sample MC simulation. A point (x, y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate). A 45 degree reference line is also plotted.

Figure 5.3 shows that QMC with 50 samples or more is good enough to say it has a similar distribution to 10,000-sample MC, while this is not the case for QMC with 25 samples. All the experimental results suggest that, a QMC sample size equal to 25 is not good enough for the statistical analysis of the folded-cascode-amplifier, whereas 50 samples seem acceptable.

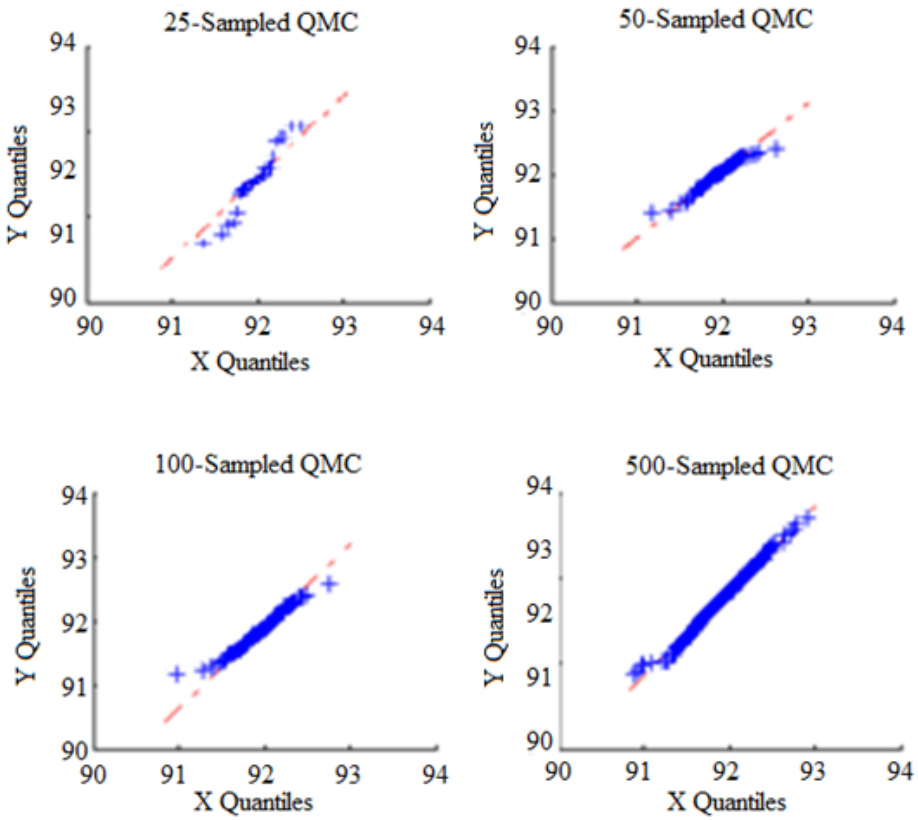

Figure 5.3. QQ-Plot of Several QMC Sample Sizes Compared to 10,000 MC Simulation [17]

For the two-dimensional optimization of the folded-cascode amplifier, the selected QMC sample size is 50. For three-dimensional yield-aware optimization of the folded-cascode amplifier, on the other hand, a linear increment of the QMC sample size has been applied for the yield-aware optimization iterations. The QMC sample size starts with 43 and goes up to 100 at the last iteration, with a linear increment (3 samples per generation) of the sample size per generation. This selection is based on the idea of increasing up to 100 QMC samples at the very last iteration, where the final yield-aware PF is obtained. Considering that the number of yield-aware iterations, for this particular synthesis, has been selected as 20 and also it has been demonstrated that 50 QMC samples are sufficient for a coarse approach for yield estimation, the QMC sample size steps have been selected as 3, meaning 43 samples in the very first iteration.

With this methodology, replacements of the non-robust solutions with the robust ones are stimulated at the first iterations while the last iterations with higher QMC sample size guarantee the accuracy of the yield estimation [17]. The overall synthesis time is decreased by 30% compared to the case a fixed QMC sample size of 100 is used.

5.4. Synthesis Results for the Folded-Cascode Amplifier

In this section, all three yield-aware synthesis techniques have been implemented and compared, performing two- and three-dimensional optimizations of the folded-cascode amplifier.

5.4.1. Two-Dimensional Synthesis of the Amplifier

For two-dimensional optimization, a nominal optimization has been run for 75 iterations with 80 individuals in the population, followed by 15 additional iterations for the yield-aware optimization phase, employing 50 QMC samples per solution as fixed sample size. The objectives and constraints used are given in Table 5.2. Among the constraints, maximum power consumption has been set as 1mW, minimum phase margin has been set as 60 degrees and the maximum area of the transistors has been set as 10.000 μm^2. Moreover, the transistors are forced to operate in the saturation region. Matlab and Hspice running on a 1.9GHz i3-3227 processor have been used. The required time for nominal optimization with 75 iterations is 20 minutes, while the whole synthesis takes about 5 hours. In Figures 5.4 and 5.5, improvements on the yield-aware PF are shown at different iterations in order to show the convergence speed of different techniques, while Figure 5.6 gives the final PF of the different techniques with statistical simulation of the solutions. In Figure 5.6, the final PF with nominal performances and also the variations of each single design point has been shown. Tables 5.4 and 5.5 show the comparison between different techniques according to the results of the optimizations. In Table 5.4, some qualitative comparisons have been presented. The first one is the replacement speed and the qualitative comparison is based on the number of the replaced solutions (or how early the replacement of non-robust solutions with robust ones start) at the first yield-aware optimization iterations, after the nominal optimization is finished. The second one is the convergence of the final PF which basically defines the quality of the final PF formed by the nominal solutions at the last iterations for each type of techniques. The last one is the CW (cloud width) of the final PF; hence an indicator for the robustness of the solutions. This comparison has also been quantatively realized as given in Table 5.5 [17], where variation percentages of each objective and also the overall variation (vector sum of variations of different objectives) have been reported.

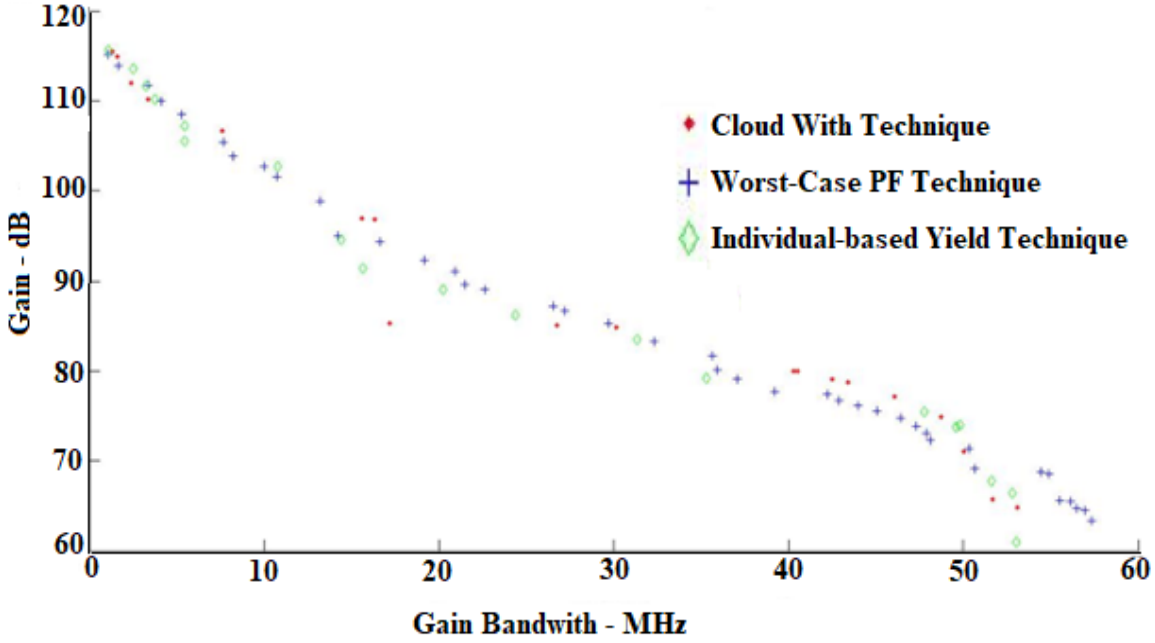

Figure 5.4. PFs obtained by Different Techniques at the 6th Iteration of the Yield-aware Optimization Phase

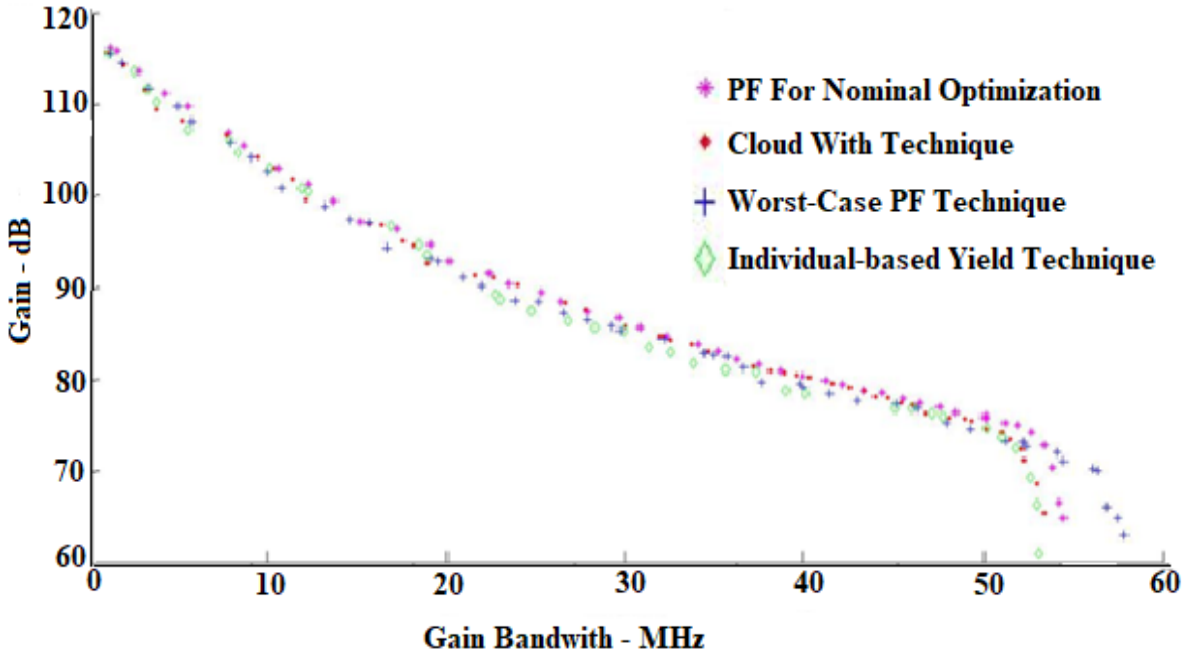

Figure 5.5. PFs Obtained by Different Techniques at the 15th Iteration of the Yield-aware Optimization Phase

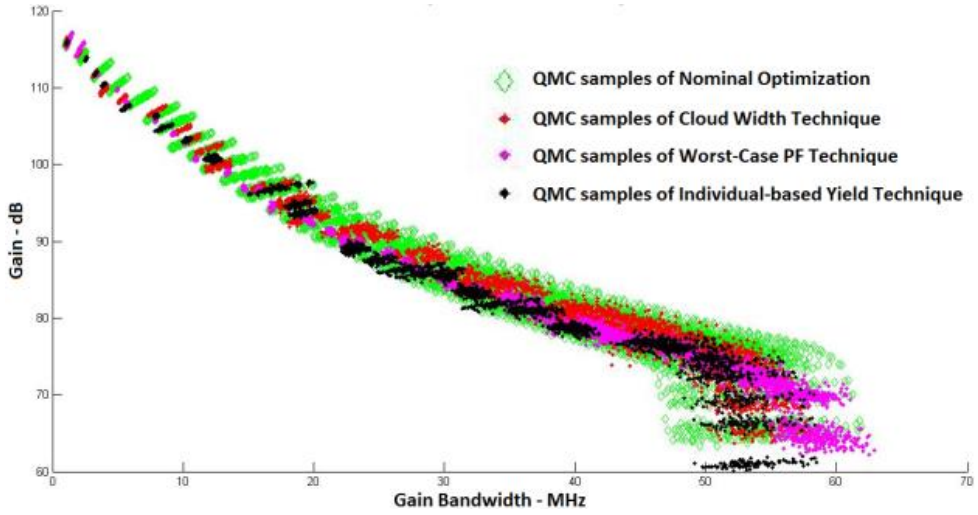

Figure 5.6. QMC Variations on the PFs for Different Techniques

Table 5.4. Performance comparison of three different yield-aware techniques

Method	**Replacement Speed**	**Convergence of the Final PF**	**Variation Width of Final PF (CW)**
CW	Good	Very Good	Good
WCPF	Good	Good	Very Good
IBY	Good	Moderate	Good

Table 5.5. Robustness comparison of three different yield-aware techniques

Method	**Objective 1 Variation (%)**	**Objective 2 Variation (%)**	**Overall Variation**
Nominal Opt.	27.4	2.6	27.5
CW	12.6	1.5	12.7
WCPF	10.3	1.8	10.5
IBY	12.4	1.6	12.5

Results show that all three techniques have decreased the variations on the final PF to approximately half of the variations of the designs on the PF of nominal optimization. It should be noted that the performances (e.g., power etc.) which were set as constraints were also checked after each variability simulation to see if the new performance values are also satisfying the initially set constraint values. If not, a penalization is applied to these solutions

to avoid the survival of the solutions with constraint violation risk and only the solutions with all variability simulations satisfying the constraints are accepted. All the solutions obtained after the nominal optimization have been replaced after 84th iteration (9 iterations after nominal optimization) for the CW and IBY techniques; while, that is the case at the 82th iteration (7 iterations after nominal optimization) for the WCPF technique. It should be noted that for such stochastical processes, this is not a meaningful difference among different approaches and all three techniques look promising for implementation with higher number of objectives [17].

5.4.2. Three-Dimensional Synthesis of the Amplifier

After the results obtained by two-dimensional optimizations of the folded-cascode amplifier; all three different yield-aware optimization techniques have been used for three-dimensional optimization problems of the same circuit. Due to the increase in the number of objectives; population size and number of iterations for both nominal and yield-aware optimizations have been increased as well. For these experiments, a population size equal to 150 has been used and Pareto fronts have been obtained after 180 iterations of nominal optimization, followed by 20 more iterations that include robustness considerations. Since the number of iterations and population size are increased compared to the two-dimensional optimization, and in order to decrease the overall synthesis time, a linear increment of the QMC sample size has been applied at the last 20 yield-aware optimization iterations. Using this method, the QMC sample size per solution starts with 43 and goes up to 100 at the last iteration, with a linear increment of 3 samples per generation. This saves approximately 30% overall synthesis time as the total number of simulations are decreased. For both three-dimensional yield-aware optimization problems, the required time for nominal optimization is 75 minutes, while the whole synthesis takes about 9 hours.

Besides the synthesis time improvement required for higher dimensional problems, a numerical metric has been utilized in order to compare the success of the proposed techniques as well. For this comparison of different yield-aware Pareto fronts generated by different techniques, the Lebesgue measure (LM) [53] has been used. The worst case objective values obtained from these PFs (same value for all calculations) have been used as a reference point for the calculations. Since the LM of a Pareto front is a metric defining

how far the solutions of the PF are from a certain reference point; results with higher Lebesgue measures are expected to have better convergence and/or diversity performance. Figure 5.7 below depicts the evolution of the Lebesgue measure of a Pareto front versus the iteration count. The Lebesgue measure saturates after the 60th iteration where this execution is not evolving further.

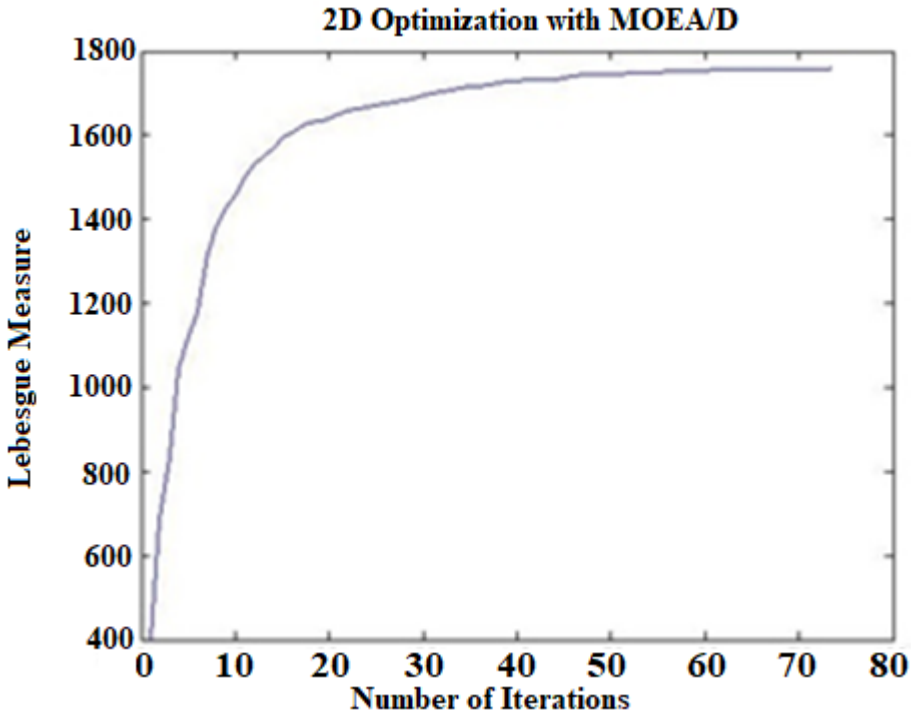

Figure 5.7. Lebesgue Measure Change with the Increasing Iterations

In order to compare the quality of the Pareto fronts using the LM metric given above, 90%-yield PFs of each solution on the overall PF have been calculated initially. To calculate a X%-yield PF from the variation space of a single solution, a tree-like search algorithm illustrated in Figure 5.8 has been implemented.

In this search algorithm, all objective values are sorted separately as given in Equations (5.1) and (5.2) for a two-dimensional optimization problem:

$$V_x = \{max_{obj1},\ \dots\ ,\ min_{obj1}\} = \{x_{10}, x_{12}, x_4, \dots, x_2\} \tag{5.1}$$

$$V_y = \{max_{obj2},\ \dots\ ,\ min_{obj2}\} = \{y_7, y_4, y_2, \dots, y_{14}\} \tag{5.2}$$

Then, the search starts with the maximum value (worst case value for a minimization problem) of one of the objectives, let us assume y_7 from V_y, followed with the sorted set V_x of the second objective; x_{10}, x_{12}, x_4 and so on, respectively, until the expected yield value is achieved. The pairs obtained in the search with the first member of V_y are as given in Equation (5.3). The yield constraint has been reached at the (x_4, y_7) pair, which is marked as the first point on the PF generated.

$$(x_{10}, y_7)\ ,\ (x_{12}, y_7)\ , (x_4, y_7) \tag{5.3}$$

After that, the search continues with the second member of the V_y as given in Equation (5.4), until another value pair with the expected yield value is achieved. The yield constraint has been reached at the (x_{12}, y_4) pair, which is marked as the second point on the PF generated.

$$(x_{10}, y_4)\ ,\ (x_{12}, y_4) \tag{5.4}$$

The whole search is finalized in that way and all marked points are kept in an external population; hence, a X%-yield PF is obtained.

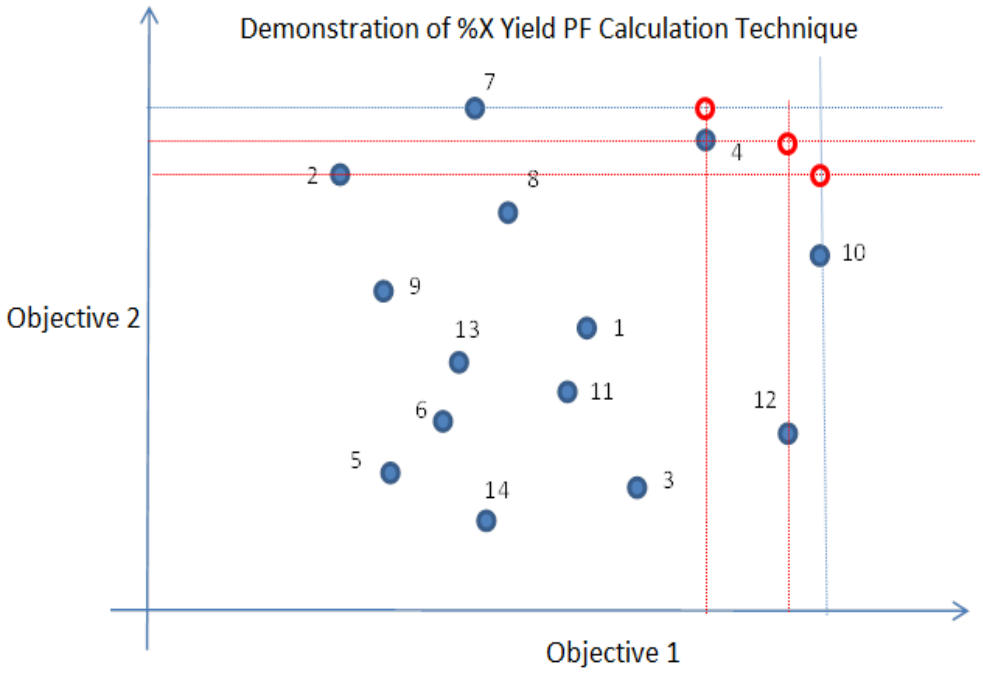

Figure 5.8. Demonstration of the Implemented Technique for 90%-yield PF Calculation

For the three dimensional synthesis results, in a yield-aware PF, 150 individuals with 100 variability samples will generate up to 15000 points on the PF. This set decreases to approximately 8000 members after the 90%-yield PF generation algorithm is applied, which is still a high value for the calculation of the hypervolume. For this purpose, in order to systematically decrease the number of samples, crowding distance technique has been applied for the 90%-yield PF. The crowding distances for the non-dominated samples of each solution is obtained by calculating the Euclidean distance of that particular sample point to a fixed reference point. Doing that, normalized performance values are used. The reference point on the other hand, is selected by using the worst performance value of each objective. These crowding distances are then sorted in decreasing order and the first 10 members of each sorted list are used to calculate the final Lebesgue measures.

The Lebesgue measures of the 90%-yield Pareto fronts of a first three dimensional optimization involving gain-bandwidth product, dc gain and phase margin as objectives are shown in Table 5.6. The table shows the results for the three techniques previously described and compares them with the results of the nominal optimization after the first 180 iterations. MOEA/D being an stochastic algorithm, the table shows the results of three different executions and the mean value of the results.

The results of a second three-dimensional experiment are shown in Table 5.7. Gain-bandwidth product, dc gain and power consumtpion were selected as objectives in this case. Agressive optimization constraints were set, e.g, the minimum overdrive voltage of transistors was set to 0 and the minimum transistor lengths were set to the minimum allowed by the technology: 0.35μm, so that performances could be optimized while, at the same time, the yield-aware optimization process guarantee that the solutions are robust enough. In order to compare these results with typical designers' approaches, they were emulated by running a nominal optimization with conservative margins aimed at obtanining robust designs: the minimum overdrive voltage was set to 200mV and the minimum transistor length to 1μm. This technique has been called the emulation of a designer's approach.

The comparative results of the Lebesgue measure of several executions are shown in Table 5.7. Tables 5.8 and 5.9 show the worst case values of a 100-sample QMC for two different solutions from the PFs comparing the CW-based yield-aware synthesizer output

and the emulation of a designer's approach. These points are selected due to the close performance values of the GBW and the gain.

Table 5.6. Lebesgue Measure of different techniques for the first experiments on 3D synthesis

Method	Test-1	Test-2	Test-3	Mean	Rank
CW	0.4457	0.4477	0.4421	0.4452	1
WCPF	0.4375	0.4355	0.4382	0.4371	2
IBY	0.4041	0.3998	0.4022	0.4020	3
Nominal Opt. @ Iteration 180	0.3754	0.3706	0.3742	0.3234	4

Table 5.7. Lebesgue Measure of different techniques for the second experiments on 3D synthesis [17]

Method	Test-1	Test-2	Test-3	Mean	Rank
CW	0.3991	0.4002	0.3980	0.3991	1
WCPF	0.3972	0.3955	0.3966	0.3964	2
IBY	0.3586	0.3544	0.3578	0.3569	4
Nominal Opt. @ Iteration 180	0.3276	0.3234	0.3244	0.3251	5
Nominal Opt. with Emulation of Designer's Criteria	0.3774	0.3831	0.3808	0.3804	3

Table 5.8. First solution comparing the results obtained from CW-based yield-aware optimization and the emulation of a designer's approach

Method	GBW [MHz]	Gain [dB]	Power [mW]	All Constraints Met?
CW	14.21	74.38	0.3455	Yes
Emulation of a Designer's Approach	14.22	74.28	0.9094	Yes

Table 5.9. Second solution comparing the results obtained from CW-based yield-aware optimization and the emulation of a designer's approach

Method	GBW [MHz]	Gain [dB]	Power [mW]	All Constraints Met?
CW	27.67	70.99	0.7955	Yes
Emulation of a Designer's Approach	26.83	66.95	1.7962	Yes

Results show that, all three different techniques dramatically improve the yield performance of the final PF compared to the nominal optimization phase of the synhthesis. These three techniques were comparable for two-dimensional optimization, however, according to the three dimensional synthesis results, for all different optimization settings and different runs, CW and WCPF techniques are much better than the IBY technique. CW and WCPF based yield-aware optimization results are also better than the synthesis results that were realized with the emulation of a designer's criteria. Even though the optimization realized with the constraints of a designer are competitive with CW and WCPF in terms of robustness quality of the final PFs, several performance values like GBW and DC Gain can only be achieved with higher power, meaning it has worse optimization performance in terms of the objectives. Overall results suggest that both CW and WCPF techniques are better than IBY technique and the designer's approach optimization; while CW has the first rank for all different problems and runs.

6. YIELD-AWARE OPTIMIZATION OF A CAPACITIVE MEMS ACCELEROMETER SYSTEM

The second part for the evaluation of the implemented algorithms is based on the synthesis of a MEMS accelerometer system. For that purpose, initially, a MEMS accelerometer sensor topology has been selected and the accuracy of the analytical models developed for this type of devices has been confirmed by comparing the model data with the results of the simulations performed using a dedicated MEMS simulator of the commercial Comsol software [54]. Running iterative MEMS simulations within an optimization loop becomes prohibitive since a single simulation takes typically around 10 minutes. Considering that the number of simulations will be *N.I*, where *N* is the population size and *I* is the number of iterations, the overall time required for MEMS simulations is too large. Hence, analytical models have been used for the evaluation of the sensor performance in the optimization loops. This chapter includes the experiments performed for the nominal optimization of the MEMS accelerometer sensor, the co-optimization of the MEMS sensor and the read-out circuitry for optimal mixed-domain system performance and, finally, the yield-aware optimization for the complete accelerometer system.

6.1. Nominal Optimization of the Capacitive MEMS Accelerometer Sensor

Before setting up the optimization for the complete MEMS accelerometer system, multi-objective optimization of a MEMS sensor with a pre-selected topology is performed.

6.1.1. The Capacitive MEMS Accelerometer Sensor

For the optimization of the MEMS sensor, several optimization variables can be used. These design variables correspond either to the topology-related parameters like the number of fingers in a comb-finger based MEMS topology, or to the feature sizes of a fixed topology. For our work, the topology selection is based on the toolset constraints of the fabrication environment that allows the fabrication of MEMS devices which are using wafer bonding. Hence, the sensor given in Figure 6.1 has a convenient process flow for the

fabrication cleanroom. The manufacturing of the MEMS includes several steps like wafer-to-wafer bonding, vacuum packaging and deep silicon etching which fits well with the tool set of the manufacturing environment. Figure 6.2 shows the moving mass, which is the middle electrode of the capacitive MEMS accelerometer sensor [10].

The MEMS sensor topology has one moving and two fixed electrodes. Two different gaps at the top and bottom sides create a differential capacitance pair with respect to the acceleration direction; hence, a voltage signal can be generated from these capacitance changes [6].

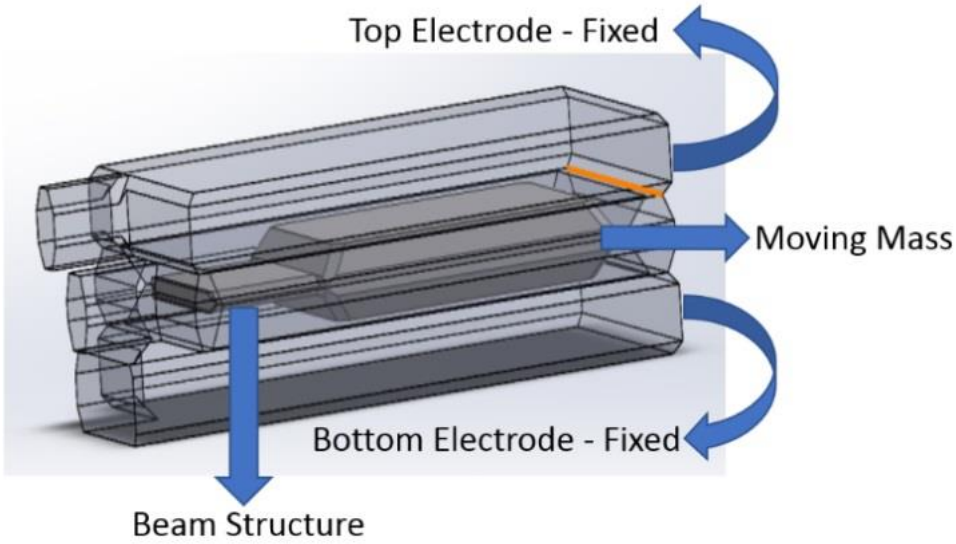

Figure 6.1. The Capacitive MEMS Accelerometer Sensor [6]

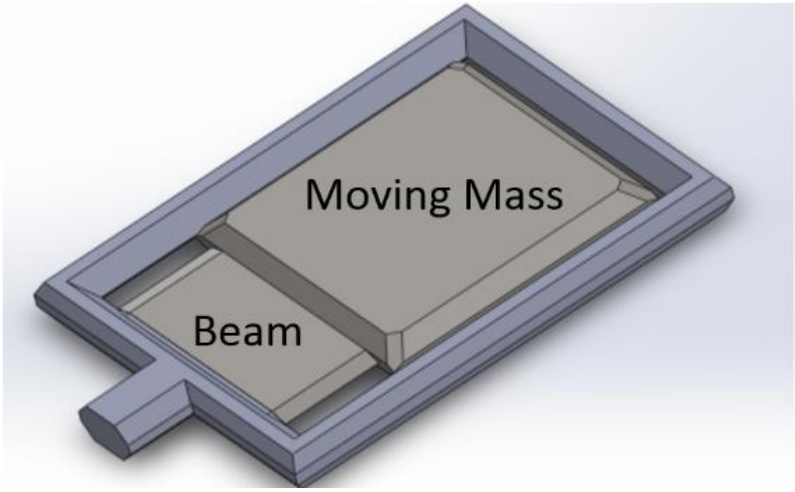

Figure 6.2. The Moving Mass of the MEMS Device [6]

6.1.2. Sensor Models

It is simply not feasible to run MEMS device simulations within an iterative optimization loop. The reason is that a single device simulation (using e.g., the MEMS module of the multiphysics simulation package Comsol) [54] takes around ten minutes. Considering that the number of simulations in a typical evolutionary optimization algorithm is high, it results in an impractical computation time using device simulations. To solve this problem, highly accurate, well-known analytical models (under vacuum conditions) can be used [55-58].

From the analytical model point of view, for the determination of the spring constant, the displacement on the spring side of the mass (per *a* accelaration) has to be initially calculated. As depicted Figure 6.1, the mass is carried by a single cantilever beam and the entire load is at the free end of the beam. The deflection at the free end of the beam in such topologies can be calculated with the analytical model [56-58] given below:

$$disp1 = \frac{4.m.a.{L_{beam}}^3}{W_{beam}.{t_{beam}}^3.E} \tag{6.1}$$

where $disp1$ is the displacement at the free end of the beam, *m* is the overall mass carried by the beam, *a* is the acceleration applied, *E* is the Young's modulus of the material used, W_{beam}, L_{beam} and t_{beam} are the width, length and thickness of the sensor beam.

After the calculation of the displacement, the spring constant of the beam structure can be calculated as:

$$k = \frac{m \cdot a}{disp1} \tag{6.2}$$

By using the spring constant and the mass, the resonant frequency can be calculated as given in [55]:

$$\omega_r = \sqrt{\frac{k}{m}} \tag{6.3}$$

The analytical model of the Brownian sensor noise [55], caused by the random collision of air molecules with the sensor, is:

$$\sqrt{\frac{a^2}{\Delta f}} = \frac{\sqrt{4.k_b.T.\omega_r}}{m.Q} \tag{6.4}$$

Equation (6.4) shows that the noise is a function of the temperature T, the resonant frequency of the device ω_r, the Boltzmann constant k_b, the overall system mass m and the quality factor Q, which is determined by the damping level of the system, hence, the vacuum packaging technology used. Selecting $Q = 500$ is a practical assumption since most of the packaging tools can achieve such good vacuum levels [6].

The last equation evaluates the bending angle of the spring mass system, θ, to calculate the displacement at the other side of the mass, which allows the calculation of both top and bottom capacitances for the moving mass structure. Calculation of the bending angle is based on the small angle approximation [56] as the displacement itself is quite small compared to the length of the beam structure:

$$\theta = \frac{disp1}{L_{beam}} \tag{6.5}$$

The accuracy of the noise model has been verified using two different commercial sensors with the same topology and material properties that have been used for all the experiments in this work. Both devices have been simulated using the Comsol Multiphysics MEMS Module, with stationary type of study, and the simulation results for the displacement of the moving mass have been used to initially calculate the spring constant in Equation (6.2), and the resonant frequency in Equation (6.3), and, finally, the noise using the analytical model in Equation (6.4).

The output of the Comsol simulation is the displacement per acceleration as given in Figure 6.3. Figure 6.4, on the other hand, shows the stress level at the moving mass surface.

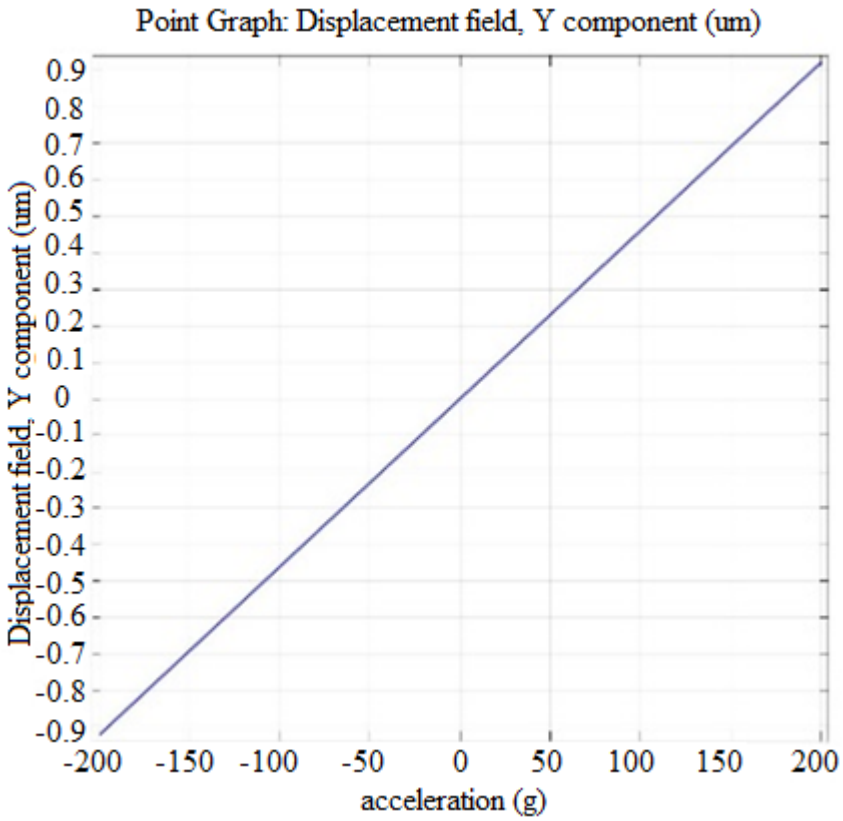

Figure 6.3. Displacement vs g Acceleration [10]

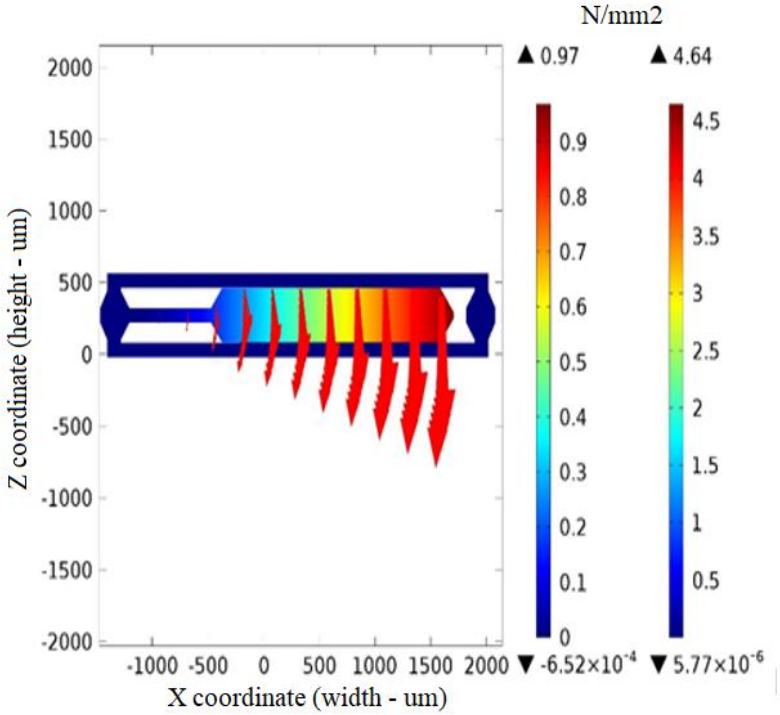

Figure 6.4. Stress Level of the Surface [10]

Comparisons between the analytical model of the noise and the noise values obtained from the device datasheets (which refer to actual analog noise measurements) are shown in Table 6.1.

The analytical model errors for the two commercial devices are 1.3% and 1.8%. The corresponding spring constant errors are 1.7% and 3.3%, respectively. The actual values are given in Table 6.2.

Table 6.1. Calculated and datasheet noise for the commercial devices

Device	**Analytical Model**	**Datasheet**
Commercial #1	18.2 ($\mu g/\sqrt{Hz}$)	18.0 ($\mu g/\sqrt{Hz}$)
Commercial #2	12.6 ($\mu g/\sqrt{Hz}$)	12.4 ($\mu g/\sqrt{Hz}$)

Table 6.2. Calculated and simulated spring constants using displacement results

Device	**Simulated**	**Analytical Model**
Commercial #1	7.3 (kN/m)	7.4 (kN/m)
Commercial #2	15.4 (kN/m)	14.9 (kN/m)

Thirty-two different device simulations with Comsol, using a uniformly distributed set of design variables, have been performed as additional verification of the accuracy of the spring constant model. The spring constant model, as given in Equation (6.2), is a function of all three dimensions of both the beam and the mass structures. Initially, 12 corner simulations, setting up the minimum and maximum values of these 6 design variables as well as the exact dimensions of the two commercial sensors were performed. Another 18 simulations with randomly selected dimensions were carried out in order to cover the entire range of the design parameters considered. The main reason for running these extra verification simulations is that the evolutionary operators in the optimization loop might come up with some solutions (design points) whose dimensions are quite different from those of the commercial devices. Displacements with respect to the acceleration have been obtained via Comsol simulations and the same displacements have been calculated using the analytical model. Figure 6.3 shows all 32 spring constant values obtained by Comsol

simulations and the analytical model while Table 6.3 shows the minimum, maximum and mean errors of the 32 MEMS device evaluations [6].

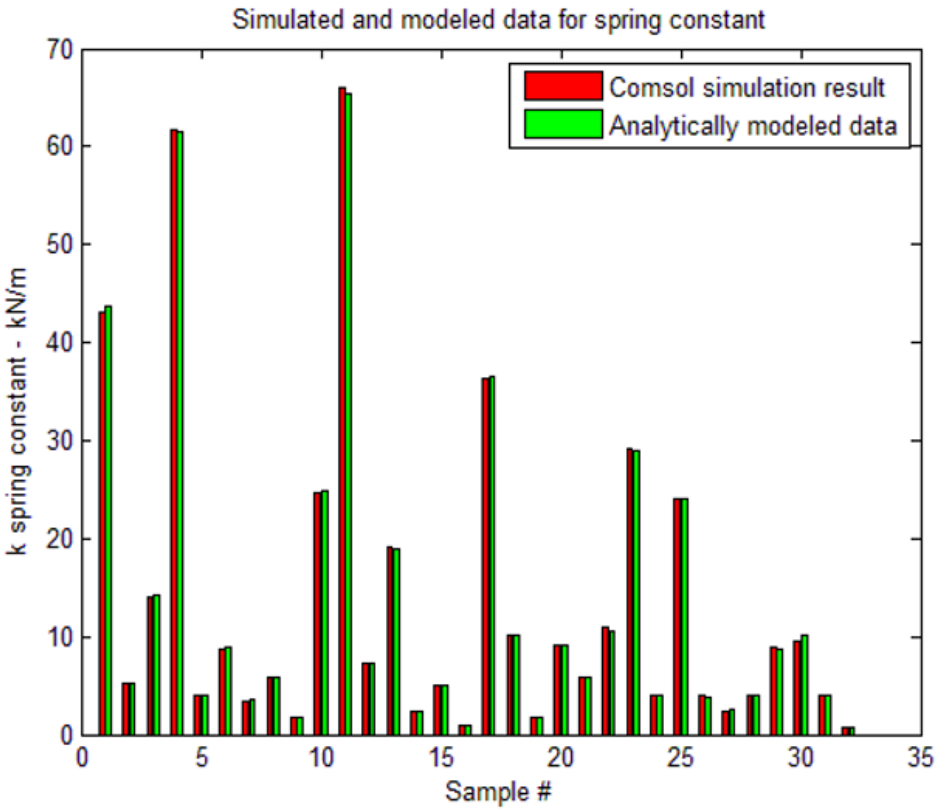

Figure 6.5. Simulated and Modelled Data for the Spring Constant

Table 6.3. Analytical model error for the spring constant

Mean Error	Maximum Error	Minimum Error
1.3%	6.5%	0.1%

6.1.3. Optimization Results for Two-Dimensional Nominal Optimization of the Sensor

The analytical model of the sensor has been integrated within the MOEA/D optimization loop, for the evaluation of objectives and constraints. Optimization settings are handled by selecting the design variables and also defining the objectives and constraints of the optimization. The synthesis loop for the capacitive MEMS accelerometer sensor has been set up for two-dimensional optimization where the objectives to be minimized are the sensor

area and the sensor noise. The accuracy of the noise model has been validated in Table 6.1. The area, on the other hand, is an objective calculated with 100% accuracy since it is just a function of the sensor dimensions. In order to calculate both performance parameters, all three dimensional data of both the beam and the moving mass are required. Beside these two objectives, the measurement range for the sensor has been set as a constraint, which is equal to or above 100 gravity, or g. This parameter is related to the movement capabilities of the mass structure among the capacitive gaps. In other words, the manufacturing conditions allow moving safely (no material sticking, etc.) for 70% of the capacitive gaps. In order to adapt that constraint to the optimization loop, the capacitive gap has been considered as an additional design variable. Table 6.4 summarizes the design parameters used as the optimization variables during the sensor synthesis [10].

Table 6.4. The design variables used during the optimization

Design Variable	**The search range**
Length of the mass	200 µm – 3 mm
Width of the mass	200 µm – 3 mm
Thickness of the mass	250 µm – 500 µm
Length of the beam	200 µm – 2 mm
Width of the beam	200 µm – 3 mm
Thickness of the beam	50 µm – 150 µm
Capacitive gap	0.5 µm – 1.5 µm

Capacitive change per g acceleration has also been used as a constraint (equal to or above 100fF) in order to relax the specifications of the input stage of a C/V converter that will, in a MEMS accelerometer system, convert the capacitance change to a voltage signal. This calculation is based on the area of the moving mass surface and the capacitive gap, hence no extra design variables are required.

Following the integration of the analytical models of the capacitive MEMS accelerometer sensor as a performance evaluator within the MOEA/D optimizer, the algorithm parameters have been set in order to run the two-dimensional sensor optimization. These parameters are the number of generations, which has been set as 150 and the population size, which has been set as 100 with a niche of 30.

Table 6.5 shows the objectives and constraints used for the two-dimensional optimization of the sensor.

Table 6.5. Objectives and constraints for the two-dimensional sensor synthesis

Performance	**Objective/Constraint**
Sensor Area	Minimize (Objective)
Sensor Noise	Minimize (Objective)
Maximum Allowable Measurement Range	> 100g (Constraint)
Capacitive Change per g Acceleration	> 100fF (Constraint)

The results have been compared with a commercial device with the same sensor topology and the same type of material used for the fabrication of the device. For a fair comparison, the constraints have been set equal to the real performance outputs of the commercial device compared. Hence, besides the area and the noise performances to be compared, the maximum allowable measurement range and the capacitive change per g values are guaranteed to be equal to or typically better than the commercial device for each single design point on the Pareto Fronts obtained.

The optimization has been run twice to guarantee the consistency of the outputs and to have more data for comparison with the commercial design. Overall time for each optimization loop is around 15 minutes with a 1.9 GHz i3-3227 microprocessor. The two Pareto Fronts obtained are given in Figure 6.6. The design point of the commercial device is also shown with an area of $6.04mm^2$ and $18.00\mu g/\sqrt{Hz}$ on the same plot as the Pareto Fronts. The design point of the commercial device is directly used from the datasheet.

Results show well distributed Pareto Fronts with remarkable improvement on the device performance compared to the commercial device specifications.

For comparison of the results with the commercial design, two different solutions have been extracted from the Pareto Front. The constraints imply equal or better performance than the commercial design for the measurement range and the capacitive change per g acceleration. The selection of design point #1 is intended for the comparison of the noise

performances for a similar area. Design point #2 on the other hand, has been selected for the comparison of the area performances for a similar sensor noise.

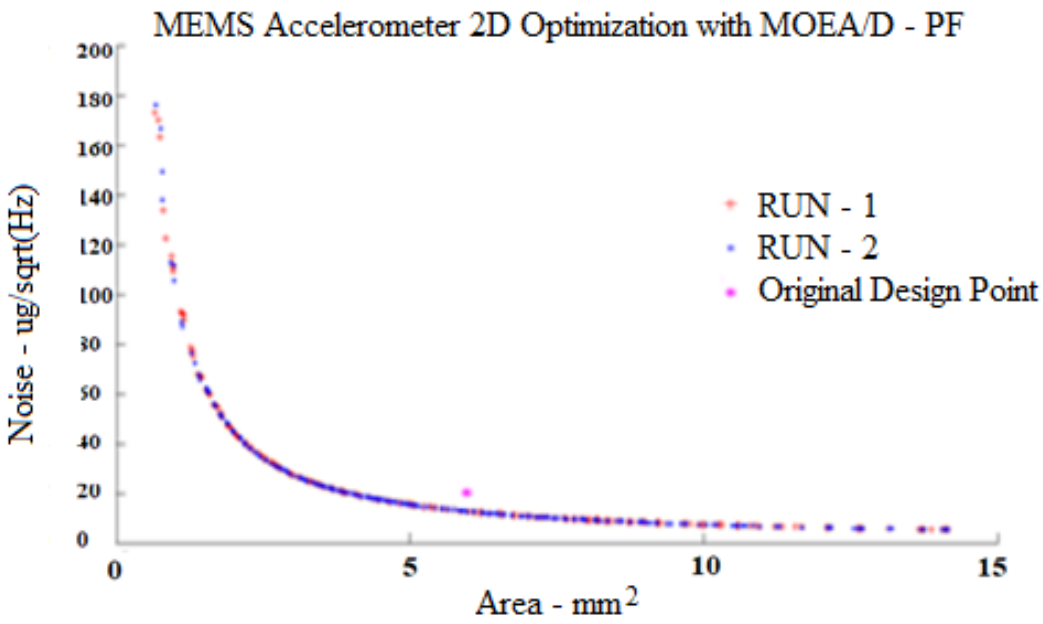

Figure 6.6. Two-dimensional PF for the MEMS Accelerometer Sensor Synthesized

The two extracted sensor designs and the commercial design have been simulated using Comsol in order to have a fair comparison and also to guarantee the accuracy of the final results. The performance values of the commercial design and two different designs are given in Table 6.6 with respect to the final Comsol simulations [10].

Table 6.6. Performance values for the commercial design and two different solutions extracted from the Pareto Fronts

Objective	**Commercial**	**Design Point #1**	**Design Point #2**
Area	6.04 mm^2	6.06 mm^2	4.54 mm^2
Noise	18.00 $\mu g/\sqrt{Hz}$	12.49 $\mu g/\sqrt{Hz}$	17.70 $\mu g/\sqrt{Hz}$

Comparing the Comsol simulation outputs of two PF-extracted design points and the commercial design, 29% better noise performance is achieved for a similar sensor area. The second design point shows a similar noise performance that can be obtained in 25% smaller sensor area.

The results in Table 6.7, on the other hand, show the accuracy of the models. These results correspond to the deviations between the analytical model and the Comsol simulation outputs for the spring constant. Noise, on the other hand, has been calculated with the same analytical model using the spring constant, in order to see the error percentages induced by the spring constant calculation [10].

Table 6.7. Analytical model accuracy for two different design points and the commercial device

Device	Spring Constant Error (%)	Noise Model Error (%)
Design Point #1	1.8	0.9
Design Point #2	1.3	0.65
Commercial	0.8	0.4

Table 6.8 compares the dimensions of the Design Point #2 with the commercial design. The second design point has a similar noise performance with a much smaller area compared to the commercial design.

As the dimensions given in Table 6.8 show, the width of the mass has been set to a quite small value by the optimization algorithm, compared to the commercial design. This has a dominant effect on the area performance. On other hand, some parameters like the thickness of the mass have been set to higher values to lower the Brownian noise value. The optimization process evolves to a gap value higher than the commercial design, in order to satisfy the constraints mentioned above.

Table 6.8. The design point #2 dimensions compared with the commercial design

Parameter	Commercial Design	Design Point #2	Difference (%)
L – Mass	2200 µm	2597 µm	18% higher
W – Mass	2000 µm	1187 µm	41% smaller
t - Mass	380 µm	500 µm	31% higher
L – Beam	820 µm	1233 µm	50% higher
W – Beam	1510 µm	1177 µm	22% smaller
t – Beam	100 µm	105 µm	5% higher
Gap	1 µm	1.5 µm	50% higher

6.2. Nominal Optimization of the Capacitive MEMS Accelerometer System

After describing the optimization of the MEMS sensor based on analytical model evaluations, a novel methodology that realizes the co-optimization of the MEMS sensor and a transimpedance amplifier-based read-out circuitry has been proposed. To enable the mixed-domain (mechanical sensor and electrical read-out circuitry) optimization, the MEMS sensor has been modelled with an equivalent circuit that consists of two capacitances calculated using the sensor model, as well as the sensor noise at the input of the read-out circuitry.

6.2.1. The Capacitive MEMS Accelerometer System and the Design Variables

Figure 6.7 shows the equivalent electrical circuit of the sensor. The topology selected for the capacitance-to-voltage (C/V) converter to measure the capacitive changes on the MEMS capacitive electrodes, on the other hand, is a conventional transimpedance amplifier (TIA)-based converter with a resistance and capacitance feedback. The system, including the MEMS element and the amplifier, is given in Figure 6.8. The capacitive change at the MEMS sensor creates a sensing current which is converted into a voltage value at the output of the amplifier. The amplifier used for the read-out circuitry is in folded-cascode configuration and was shown in Figure 2.5.

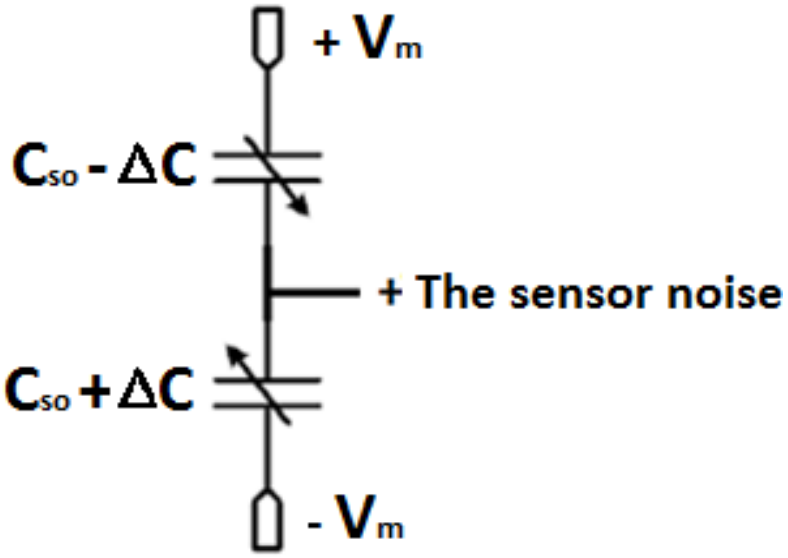

Figure 6.7. The Equivalent Circuit Used for the MEMS Accelerometer Sensor [6]

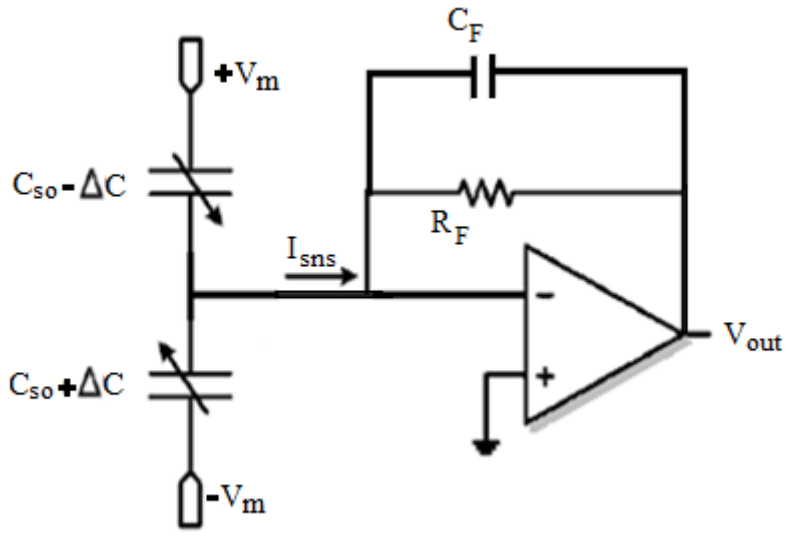

Figure 6.8. MEMS Accelerometer with the TIA-based C/V Converter [6]

In order to synthesize the MEMS accelerometer, several design variables of the different building blocks have been used. These design variables are the three dimensions (thickness, width and length) of the mass and the beam structures of the sensor, the capacitive gap between the moving mass and the fixed electrodes, the bias current, the transistor widths and length values of the amplifier and finally the feedback resistor and capacitor values of the C/V converter. The allowable ranges for the design variables are given in Table 6.9.

Table 6.9. Allowable ranges of the design variables

Design Variable	**Minimum Value**	**Maximum Value**
$L_{mass}, W_{mass}, W_{beam}$	200µm	3mm
t_{mass}	250µm	500µm
L_{beam}	200µm	2mm
t_{beam}	50µm	150µm
C_{gap}	0.5µm	1.5µm
$W_{transistors}$	0.24µm	100µm
$L_{transistors}$	0.18µm	10µm
R_F	10kΩ	1MΩ
C_F	0.1pF	10pF
I_{bias}	0.5µA	2.5mA

6.2.2. The Proposed Co-Optimization Technique for the Accelerometer System

In order to implement an automated design methodology for the mixed-domain accelerometer, a conventional approach is the hierarchical top-down optimization of the MEMS. In hierarchical top-down optimization, the system level synthesis is decomposed into smaller optimization subproblems like the sensor and the input stage electronics. The main advantage of this technique is the smaller size of the search space of each subproblem, easing the convergence of optimization methods. On the other hand, the need for the partitioning of some common specifications (e.g., the system noise which is contributed by both the electronic noise and the sensor noise) between both subproblems makes it more challenging, especially for mixed-domain systems like MEMS [6].

The alternative approach proposed in this work consists of a co-optimization process of the whole MEMS, where the sensor performance is evaluated first at each iteration of the optimization loop, and, then, the sensor evaluation results are used for circuit level simulations, both performed within the same optimization loop. The optimal performances are obtained by MOEA/D. The proposed technique is expected to bring solutions to the specification partitioning problem since such partitioning disappears from the design process. It should also be noticed that thanks to the co-optimization, problems related to composition of separately optimized sensor and circuitry are avoided.

The methodology proposed requires evaluations at both, the circuit and the sensor level. At the circuit level, simulations are performed using Hspice, while the MEMS sensor performance evaluations are realized using analytical models. The evaluation flow of the proposed methodology is illustrated in Figure 6.9. In order to demonstrate the proposed technique, the MEMS accelerometer system given in Figure 6.8 has been synthesized. Figure 6.10 shows the flow diagram of a single iteration of the proposed optimization methodology for this mixed-domain MEMS accelerometer.

In a single iteration of the optimization loop, initially, the system level design variables are generated by the optimization algorithm. Later, the MEMS analytical model is evaluated using the values of the design variables which are dedicated to the MEMS sensor. The performance outputs such as area (to calculate the contribution of the sensor to the total

manufacturing cost) and noise are extracted from the evaluations of the MEMS model. The model is also used to obtain the capacitive interface (bottom and top capacitances per *g* acceleration) of the MEMS sensor to be adapted to the Hspice netlist of the read-out circuitry, together with the sensor noise to be adapted as a noise source at the amplifier input, in order to simulate the system level performance outputs. The design variables dedicated to the electrical simulations, which are the design variables of the amplifier and the feedback components, are introduced at this level as well. Once the system level performance values are obtained, the optimization algorithm uses these outputs, as well as the design variables providing these performances, in order to improve the quality of the system design in the next iteration.

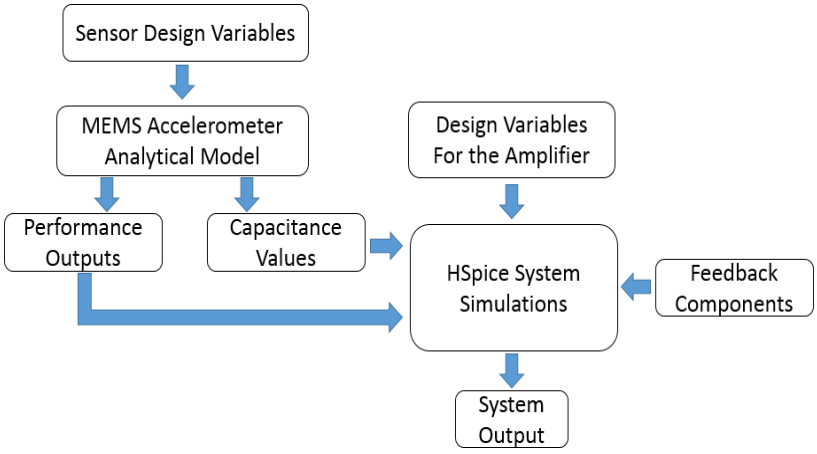

Figure 6.9. Evaluation Flow Used for MEMS Accelerometer System Synthesis

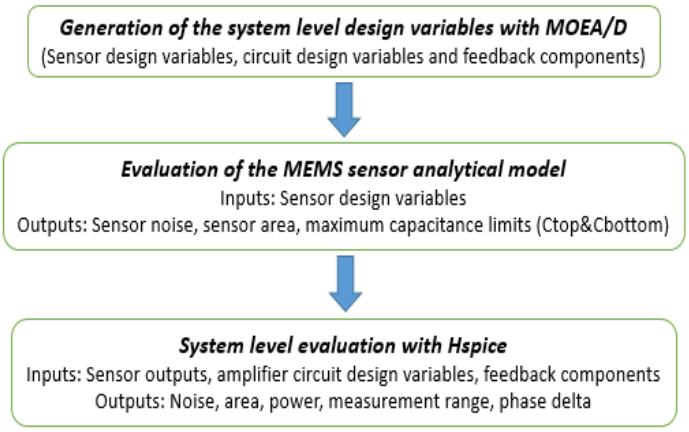

Figure 6.10. Evaluation of a Single Iteration in MEMS Accelerometer Synthesis Using the Proposed Methodology

From the circuit level simulations, the system level outputs, i.e., the measurement range, phase delta, system noise, total system cost and the power consumption, are obtained. Depending on the user selection, these system performances can be selected as either objectives or constraints.

6.2.3. Comparison of the Proposed Technique with Conventional Top-Down Approaches

The co-optimization of the MEMS accelerometer is expected to avoid some significant drawbacks of alternative design methodologies. In particular, the co-optimization of the sensor and the circuit is expected to result in optimal (and indirect) partitioning of the MEMS specifications between both the circuit and the sensor. To demonstrate this, the proposed methodology is compared with a top-down design methodology in which the sensor is optimized first, and, then, the circuitry is optimized for this optimal sensor. This philosophy emulates some reported approaches e.g., that used in [14]. In this methodology, the user-defined partitioning of specifications between sensor and surrounding circuitry may result in suboptimal system design, even though the sensor and the circuit can be individually optimal. Let us consider the single-objective optimization of the MEMS accelerometer system. The optimization was run with a population of 100 individuals with 100 iterations [6].

Table 6.10 shows the system level specifications. The total manufacturing cost of the MEMS system is the minimization objective while the other performance outputs are handled as constraints. The total system cost in USD is given by:

$$total_{cost} = 0.58 \cdot \left(Area_{tr} + Area_{fb}\right) + 0.16 \cdot Area_{MEMS} + 2 \tag{6.6}$$

The equation above assumes a hybrid manufacturing approach in which the MEMS sensor and the circuit including the feedback components are manufactured separately and, then, bonded. Hence, different coefficients, determined by the manufacturing costs per area for each related technology, appear in the total manufacturing cost calculation.

The coefficients 58 $cents/mm^2$ and 16 $cents/mm^2$, on the other hand, were obtained from the Tubitak Turkey cleanroom facilities. These coefficients are the cost per area for standard CMOS (and the passive components) and MEMS manufacturing, respectively. The big difference between these coefficients mostly comes from the number of masks needed, which makes the complexity of the process flow for CMOS and MEMS manufacturing quite different. The additional two dollars in (6.6) correspond to the fixed packaging cost, independent of the specific MEMS design.

Table 6.10. System level specifications for single-objective optimization

Performance - System	**Value**
Overall System Cost (USD)	Minimize
Total System Noise (μg/√Hz)	< 8 / 12.5 / 25 / 50 / 80
Phase Delta (º)	< 3
Measurement range (g)	> 100
Power Consumption (μW)	< 60

For the comparison of the two different methodologies, five different optimizations for five different noise specifications, all aimed at minimizing the overall system cost, have been performed, as given in Table 6.10. The other constraints are the acceleration measurement range of the MEMS device, overall power consumption and phase delta, which is the phase difference of the system output from 90 degrees. The phase delta at the output of the C/V converter is an important quality parameter for the MEMS device since the output signal is likely to be demodulated. All the constraints, except noise, are the same for all optimizations. The limit value for the phase delta, or phase error has been selected as 3 ºs, which is a typical value in MEMS accelerometer designs [59]. The power consumption constraint, on the other hand, has been selected as 60μW. The power consumption of the read-out circuitry may span from a few hundred μWs [60] to few mWs [61]. Since the read-out circuitry used only has a single stage that works as a C/V converter, the constraint has been set to low values to enable low power designs where the circuit noise will have an impact on the system level noise as well [6].

The system objectives and constraints given above can be directly handled in the proposed co-optimization approach. However, the conventional top-down approach

optimizes the MEMS sensor first, and, then, sizes the circuit using the optimal MEMS sensor. It is obvious that cost minimization should also be an objective of both the individual sensor and the circuit optimizations. However, noise is a challenging constraint since it is introduced into the system by both the sensor and the readout circuitry. Therefore, the system noise specification in Table 6.10 must be first partitioned between the sensor and the circuit. For that purpose, experiments with five different partitionings, shown in Table 6.11, have been executed for each of the five noise specifications in Table 6.10, resulting in the formulation of 25 optimization problems in total. Practically, the contribution of the sensor noise to the overall system noise is expected to be higher than the noise contribution of the read-out circuitry. The sensor signal itself is noisy and the interface should not add an appreciable amount of noise. Hence, the circuit noise should be kept small. The choice of the partitioning values are based on this assumption.

Table 6.11. Partitioning of noise constraint among the MEMS sensor and the circuit

Partitioning	**Sensor noise**	**Circuit noise**
Partitioning #1	50%	50%
Partitioning #2	60%	40%
Partitioning #3	70%	30%
Partitioning #4	75%	25%
Partitioning #5	80%	20%

Another constraint is the minimum measurement range that the system can achieve. Both, the sensor and the circuit have direct impact on the measurement range through different mechanisms. For the MEMS sensor, the acceleration range is limited by the movement capabilities of the mass, hence, determining the capacitive gaps of the device. The movement of the mass towards the fixed electrodes might cause sticking issues. The measurement range limitation on the circuit side is a different mechanism that is determined by the supply voltage and the amplification level of the input signal. With a high level of amplification, the input signal, that was generated by the changing sensor capacitances, is likely to saturate at the system output. That may result in the impossibility to measure the acceleration values that are lower than the measurement range constraint [6].

The other constraints are the phase delta and power consumption and they are only determined by the circuit. Hence, these constraints are only applied during the optimization of the circuit that interfaces the optimal MEMS sensor. Tables 6.12 and 6.13 show the objective and the design constraints used during the isolated optimization of the MEMS sensor and circuitry, respectively, in the top-down design approach.

Table 6.12. Minimization objective and design constraints of the MEMS sensor

Performance – MEMS Sensor	**Value**
Overall System Cost (USD)	Minimize
Sensor Noise (μg/√Hz)	Depends on the partitioning
Measurement range (g)	> 100

Table 6.13. Minimization objective and design constraints of the C/V converter

Performance - Circuitry	**Value**
Overall System Cost (USD)	Minimize
Circuit Noise (μg/√Hz)	Depends on the partitioning
Phase Delta (º)	< 3
Measurement range (g)	> 100
Power Consumption (μW)	< 60

Table 6.14 shows the fabrication cost results of the proposed optimization methodology for the five different synthesis problems proposed in Table 6.10 as well as the 25 experimental results of the conventional top-down methodology for the same five synthesis problems (five different noise constraints in the first column in Table 6.14) but considering the five different partitioning solutions in Table 6.11 for each of them. It should be noticed that these results do not include the packaging cost [6].

As Table 6.14 demonstrates, the proposed methodology, as a co-optimization of the sensor and the circuit, yields better results than the alternative top-down design methodology. Isolated optimization of the MEMS sensor first, and then the circuit optimization using the pre-optimized sensor, on the other hand, requires partitioning among some specifications like noise, for which both the MEMS and the circuit have impact on the final system performance. It can be observed that the optimal partitioning in the conventional

top-down approach is different for each noise specification. But even for the best partitioning among five different possibilities, the lowest system cost obtained is higher than the proposed optimization technique for all five different system noise specifications.

Table 6.14. Single-objective optimization results of the fabrication cost (in USD) for the proposed optimization methodology and the conventional top-down design approach

Noise Constraint (μg / $\sqrt{Hz}$)	**Cost Proposed Approach**	**Cost Partitioning #1**	**Cost Partitioning #2**	**Cost Partitioning #3**	**Cost Partitioning #4**	**Cost Partitioning #5**
8	1.51	1.67	1.52	1.71	1.79	1.89
12.5	0.99	1.24	1.09	1.06	1.14	1.21
25	0.55	0.79	0.71	0.64	0.56	0.63
50	0.32	0.50	0.44	0.39	0.36	0.33
80	0.24	0.41	0.37	0.32	0.30	0.28

6.2.4. Optimization Results for Two-Dimensional Optimization of the Capacitive MEMS Accelerometer System

Two-dimensional optimizations of the capacitive MEMS accelerometer have been performed using the proposed synthesis approach. The overall system noise and the manufacturing cost are selected as the design objectives. Design constraints and objectives are listed in Table 6.15.

Table 6.15. System specifications for two-dimensional synthesis of the MEMS system

Performance	**Value**
Overall System Cost (USD)	Minimize
System Noise (μg/√Hz)	Minimize
Phase Delta (º)	< 3
Measurement range (g)	> 100
Power Consumption (μW)	< 300

The population size and the number of generations of the optimization algorithm are 200 and 300, respectively. The only difference of the constraints compared to the single-objective optimization is the power consumption, that was set to 300μW, as a practical

design value [60]. Figure 6.11 shows an approximation to the two-dimensional Pareto Front, which includes the solutions trading-off system noise and manufacturing cost. The snapshots for three different number of generations in Figure 6.11 enables to monitor the evolution of the PF.

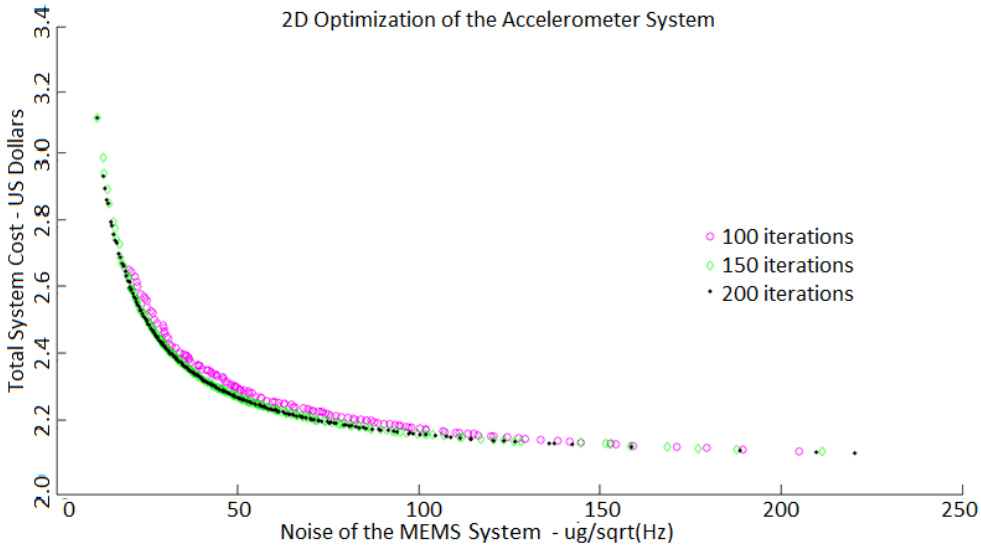

Figure 6.11. The Pareto Front of Two-dimensional Optimization

The evolution of the Pareto front shows that the optimization convergence is stalled by the 150^{th} iteration as there is no significant improvement afterwards. In order to quantitatively check the level of convergence along different iterations, the Lebesgue measure has been plotted vs. the number of iterations in Figure 6.12.

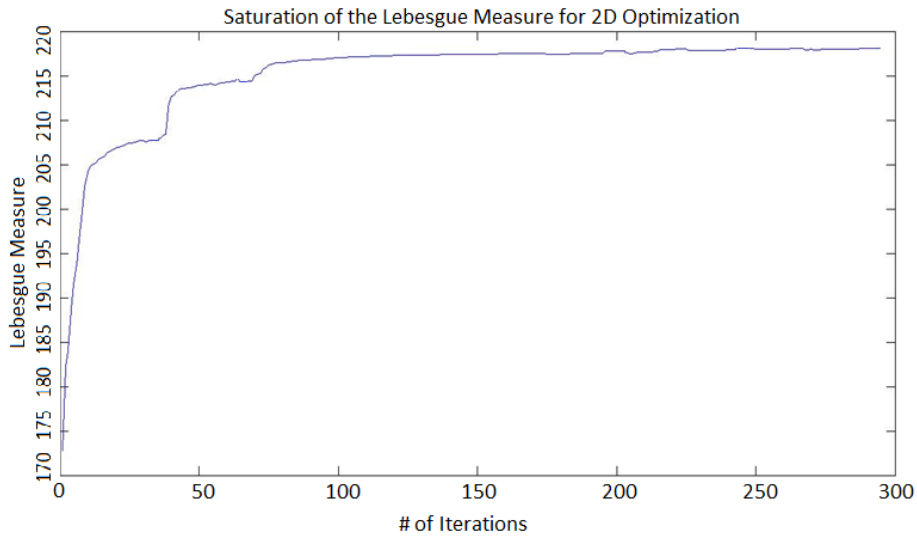

Figure 6.12. The Lebesgue Measure of Each Iteration in Two-dimensional Optimization

The results suggest that the optimization process does not significantly improve the results after around the 110^{th} iteration.

In the proposed optimization of the mixed-domain MEMS accelerometer, not only the system-level results, but also the sensor and the circuit are expected to be individually optimal. In order to illustrate that, two different PFs have been obtained. The first one is the PF obtained by applying a similar optimization approach but only focused on the MEMS sensor. The second PF is the MEMS sensor front that is obtained by extracting the MEMS sensor designs from the mixed-domain optimization of the whole MEMS accelerometer in Figure 6.11.

Figure 6.13, on the other hand, shows that the optimized MEMS sensor PF and the extracted MEMS sensor PF have very similar performances. This suggests that each optimal MEMS accelerometer design is obtained by using a MEMS sensor with some optimal trade-off among its performances.

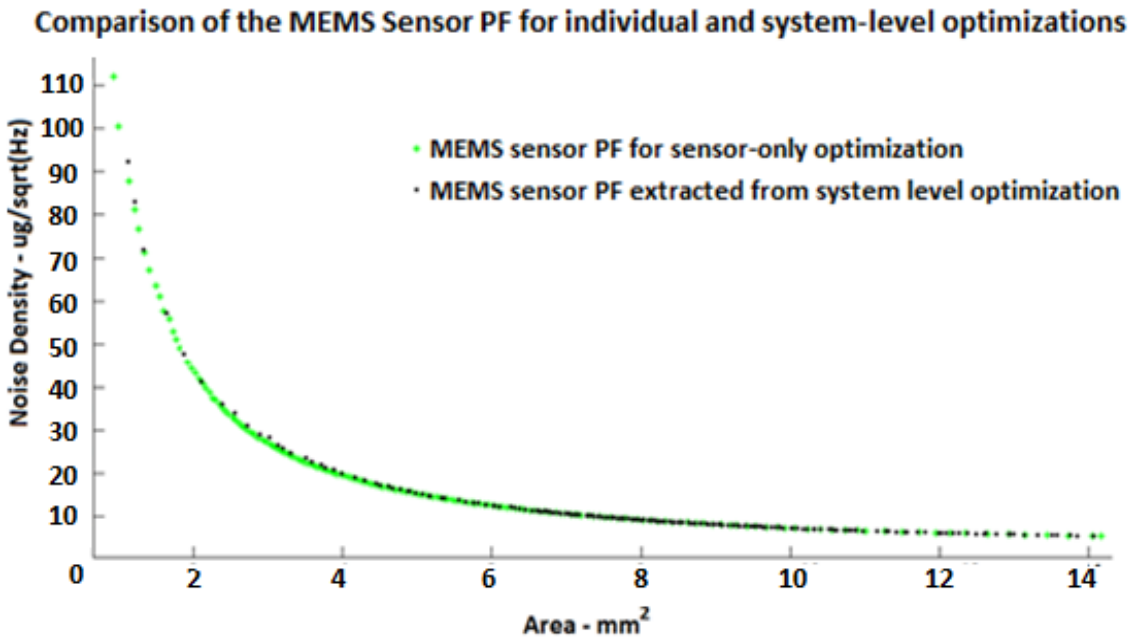

Figure 6.13. Comparison of the MEMS Sensor PF for Individual and System-level Optimizations

In order to assess the accuracy of the sensor model on the final optimization results, 11 uniformly distributed design points have been extracted from the Pareto front in Figure

6.11. The selected MEMS have been simulated using Comsol, in order to compare the simulation results with the analytically calculated spring constants. The results of the model accuracy for each MEMS design point is listed in Table 6.16. This table also shows the deviation on the MEMS system-level noise that is caused by the errors at the spring constant calculation. The highest noise deviation on the Pareto front is less than 2%. The calculation of the total system cost, on the other hand, is fully accurate since it only relies on the calculation of area.

Table 6.16. Accuracy of the spring constant and the noise for 11 different designs extracted from the MEMS PF

Design Point on PF (Solution #)	**Error of the spring constant (%)**	**Error of the MEMS Noise (%)**
1	1.65	0.82
20	3.82	1.89
40	0.88	0.44
60	1.12	0.56
80	2.80	1.39
100	2.65	1.32
120	0.45	0.22
140	1.82	0.91
160	3.04	1.51
180	2.12	1.05
200	3.66	1.81

It might seem a priori that the Pareto front in Figure 6.11 could also be obtained by composing the MEMS sensor Pareto front with the Pareto front of the readout circuit. In order to address that, Pareto fronts for the MEMS sensor and the circuitry, involving minimization of cost and noise, have been independently generated. Composing the design points from each Pareto front in order to achieve optimal system level cost and noise is, a priori, feasible. However, that might result in a constraint violation at the system level due to dependence of the TIA feedback network from the selection of the capacitive MEMS interface. Moreover, changing the values of the capacitive interface might result in a change

of the noise calculations at the circuit level, inducing a performance shift. These problems are overcome by mixed-domain co-optimization as the proposed technique performs. In order to demonstrate the risk of constraint violation and performance shift, several experiments have been performed.

In order to optimize the MEMS accelerometer with the specifications given in Table 6.15 by composing individually optimized sensor and circuit, Pareto fronts have been separately generated using the sensor specifications in Table 6.17 and the circuit specifications for a standard capacitive interface in Table 6.18. For both optimizations, manufacturing cost and noise have been set as objectives, while the constraint selection depends on the impact on the system level performance. For example, the measurement range is affected by both of the sensor and the read-out circuitry devices while the power consumption is a contribution of the electronic circuit only. The population size for sensor and circuit optimizations is 200. For both optimizations, the number of generations is 300.

Table 6.17. System specifications for two-dimensional synthesis of the MEMS sensor

Parameter / Performance	**Value**
MEMS Sensor Cost (USD)	Minimize
MEMS Noise (μg/√Hz)	Minimize
Measurement range (g)	> 100

Table 6.18. System specifications for two-dimensional synthesis of the read-out circuitry

Parameter / Performance	**Value**
Circuit Cost (USD)	Minimize
Circuit Noise (μg/√Hz)	Minimize
Phase Delta (°)	< 3
Measurement range (g)	> 100
Power Consumption (μW)	< 300

In order to demonstrate the risks of composing separately designed mixed-domain devices, 50 different MEMS sensor designs have been composed with the same optimal circuit design. The MEMS sensor Pareto front used is the front given in Figure 6.13, where 50 uniformly distributed sensor designs have been selected for their composition with a

single circuit. The selected circuit design, on the other hand, has a noise of 5 μg/√Hz and a total cost of 0.16 USD without packaging cost. One of these 50 compositions resulted in a system-level constraint violation, while five of them caused performance shift. The other 44, on the other hand have no such drawbacks but result in non-optimal design points as expected.

As the results suggest, the composition of separately designed MEMS mechanical sensor and the read-out circuitry in the electronic domain has some drawbacks that the nature of the proposed mixed-domain optimization does not have. Hence, besides the optimal partitioning advantage, the proposed optimization methodology can also overcome the problems introduced by the composition of the mixed-domain devices.

6.3. Yield-Aware Optimization of the Capacitive MEMS Accelerometer System

Process variations play an important role in the robustness of mixed-domain MEMS and should be taken into account to avoid re-spins in the fabrication. After the implementation of the novel mixed-domain co-optimization methodology for MEMS, two yield-aware optimization methodologies (CW and IBY) have been integrated into the optimization loop for the MEMS accelerometer system to develop a novel and generic yield-aware optimization methodology that can be used for mixed-domain synthesis of robust MEMS [4].

6.3.1. Multi-objective Yield-Aware Optimization Technique for the MEMS Accelerometer System

The MEMS accelerometer system given in Figure 6.8 has been used for the demonstration of the yield-aware multi-objective optimization. The evaluation of the sensor and circuit nominal performances as well as the performance variabilities are performed as in the nominal optimization methodology discussed in Section 6.2.

The optimization includes two different phases. The first phase is the nominal optimization of the MEMS accelerometer while the second one is the yield-aware

optimization by which the robustness of the pre-optimized solutions are improved. Identical to the nominal co-optimization methodology explained in Section 6.2, regardless of the optimization phase, at the begining of any iteration in the optimization loop, the sensor performance is evaluated using highly accurate analytical models in order to obtain the sensor performances. The sensor used is a capacitive MEMS sensor where the changes at the capacitance values generate the electrical signals to read-out the acceleration level. The capacitance values of the sensor are also evaluated using the analytical models and these capacitances create the capacitive interface circuit of the MEMS sensor. This capacitive interface circuit is integrated into the netlist of the read-out circuitry, together with the feedback components used in the read-out circuitry, in order to generate a system netlist. After that, system level electrical simulations are performed in order to get the MEMS performance such as the measurement range, system noise, total system cost, phase shift delta (phase difference of the system output from 90º) and the power consumption. This performance evaluation flow is shown in Figure 6.9.

As shown in Figure 6.14, at the beginning of the optimization, the algorithm runs for several iterations without including the yield concept, in order to create nominally optimized solutions before enhancing the yield.

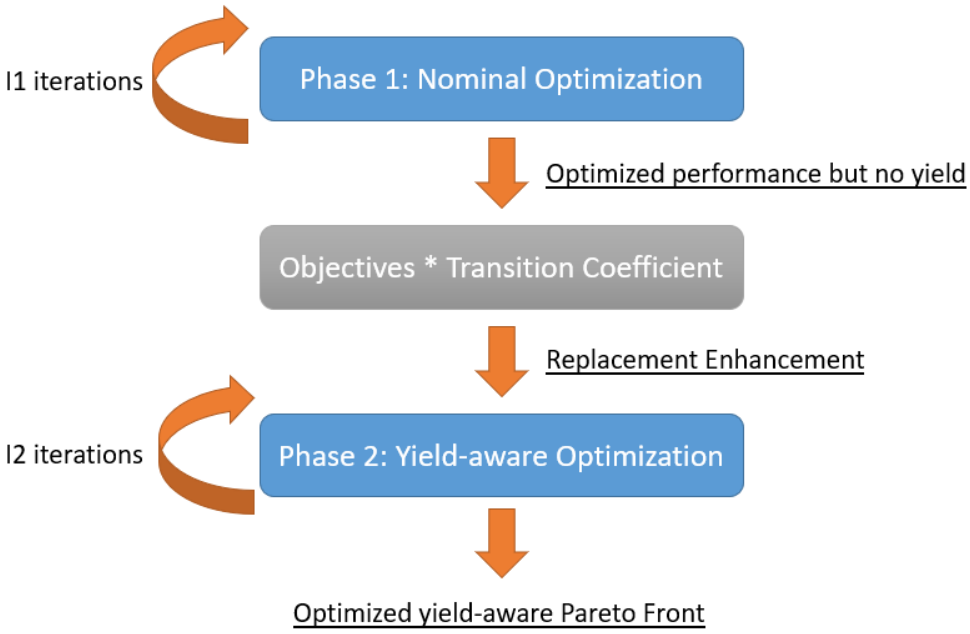

Figure 6.14. Two-step Yield-aware Optimization Methodology

Once the nominal optimization is finished, all objectives are multiplied by the transition coefficient previously defined. The MOEA/D algorithm replaces existing solutions by newly generated ones only in case there is an improvement in the performance objectives. By using the transition coefficient techniqtue in our methodology, the replacement of solutions, which have slightly worse performance than the existing ones but with improved robustness, is promoted. After the update of the objective values using the transition coefficients, one of the yield-aware optimization methodologies (CW and IBY, in this case) are run for several iterations to generate yield-aware PFs.

A dynamic stopping criterion for the nominal optimization, which is based on the Lebesgue measure has been used. In order to empirically determine the optimal transition coefficients, a statistical study based on the trade-off between the optimization efficiency and the replacement probability of the robust solutions has been carried out. The applied technique stops the nominal optimization once the increase (improvement) of the Lebesgue Measure (LM) value at a certain iteration is not higher than a certain value.

The transition coefficient, just like the stopping criterion for the nominal optimization, has been empirically determined. As mentioned in Chapter 5, the selection of the transition coefficient creates a trade-off between the convergence speed of the algorithm and the quality of the final PF. If the number is set too low, the replacements will occur quickly; however, the quality of the solutions might be poor. On the other hand, setting this number too high decreases the probability of the replacements and slows down the entire optimization process, even though the replaced solutions have good quality.

6.3.2. Selection of the Optimization Parameters

The main optimization parameters that need to be determined are the stopping criterion for nominal optimization, the transition coefficient, the QMC sample size and the constraints for the CW and the IBY methodologies. A statistical study has also been performed to enhance the replacement mechanism of the algorithm to increase the probability of the replacement of low yield solutions with the high yield ones. The parameters of this study include the number of iterations for both nominal and yield-aware optimization phases and the transition coefficients.

6.3.2.1. Statistical Study for the Replacement Mechanism Enhancement

In order to determine the optimization parameters, 20 different iteration counts and transition coefficient sets have been used to realize a two-dimensional robust optimization of the MEMS accelerometer. The population size for the optimization runs is set to 100 and the number of QMC samples is set to 50, which are two practical values whose selection is explained in more detail in the rest of the chapter. The number of iterations for the yield-aware optimization phase was set to a fixed value: 25, in order to eliminate the impact of a varying parameter on the statistical analysis experiments to determine the optimal transition coefficient. The results are shown in Table 6.19.

Table 6.19. Statistical checks for optimization parameter determination

Nominal Opt. iteration #	Yield Opt. iteration #	Transition Coef.	Opt. Time (hours)	# Robust Solutions Synthesized	LM (*10^5)
0	100	*NA*	21.33	82	2.019
50	50	0.92/1.08	11.25	88	2.098
75	50	0.92/1.08	11.29	89	2.124
95	25	0.92/1.08	5.33	96	2.167
105	25	0.92/1.08	5.35	97	2.192
115	25	0.92/1.08	5.36	54	2.193
125	25	0.92/1.08	5.36	18	2.181
95	25	0.94/1.06	5.32	97	2.18
105	25	0.94/1.06	5.34	98	2.196
115	25	0.94/1.06	5.37	41	2.195
125	25	0.94/1.06	5.39	12	2.187
95	25	0.96/1.04	5.33	98	2.189
105	**25**	**0.96/1.04**	**5.36**	**96**	**2.215**
115	25	0.96/1.04	5.36	28	2.204
125	25	0.96/1.04	5.4	2	2.194
95	25	0.98/1.02	5.33	60	2.184
105	25	0.98/1.02	5.34	54	2.187
115	25	0.98/1.02	5.36	2	2.193
125	25	0.98/1.02	5.38	0	NA
95	25	1.0/1.0	5.31	38	2.188

The LM for 1000 iterations has also been checked in order to identify how much the final PFs have converged to the ideal PF. This value is obtained for nominal optimization by averaging three different runs and is equal to $2.221 \, . \, 10^5$. Even though 1000 iterations have

been run in order to emulate the infinite number of iterations, after the 265^{th} iteration (average of three runs), no update on the PF happened as no solutions have been replaced with a better one.

It should be noted the closer any LM value gets to this value, the better the solutions. The transition coefficients are given as pairs. Considering the 0.94/1.06 pair for instance; the objectives to be minimized are multiplied by 1.06 and the objectives to be maximized are multiplied by 0.94.

6.3.2.2. Determination of the Stopping Criterion for Nominal and Yield-Aware Optimizations

It can be seen in Table 6.19 that if the number of iterations for nominal optimization is set too low, the replacement of the solutions during the robust optimization can be easily realized; however, the quality of the solutions are really low, as suggested by the LM values. On the other hand, setting this number too high decreases the probability of the replacements by so-called "freezing" the design points. Hence, the number of the iterations for the nominal optimization needs to be selected carefully. The experiments for two-dimensional optimization show that 105 iterations is a good number for the nominal optimization phase, regardless of the selection of the transition coefficient. In order to adapt a dynamic stopping criterion and define a methodology based on these experimental tests, the Lebesgue Measure has been calculated at each single iteration for both two- and three-dimensional optimizations. The stopping criterion applied is based on comparing the Lebesgue Measure at a certain iteration with the average of the past 20 iterations. If the improvement is less than 0.15%, it is considered that the improvements in the LM are marginal and running further iterations will lower the probability of robust solution replacement in the yield-aware optimization phase.

Three different runs of both two- and three-dimensional optimization have been realized according to the termination criterion based on the LM. For two-dimensional optimization, the nominal optimization has been terminated at iterations 104, 105, and 108, respectively, aligning satisfactorily with the empirically determined 105 value in Table 6.19. For three-dimensional optimization, the nominal optimization has stopped at iterations 294,

288, and 289. The LM values obtained at each iteration are given in Figure 6.15 and Figure 6.16 for two- and three-dimensional optimization, respectively. The iteration at which the stopping criterion was satisfied is also shown in the figures.

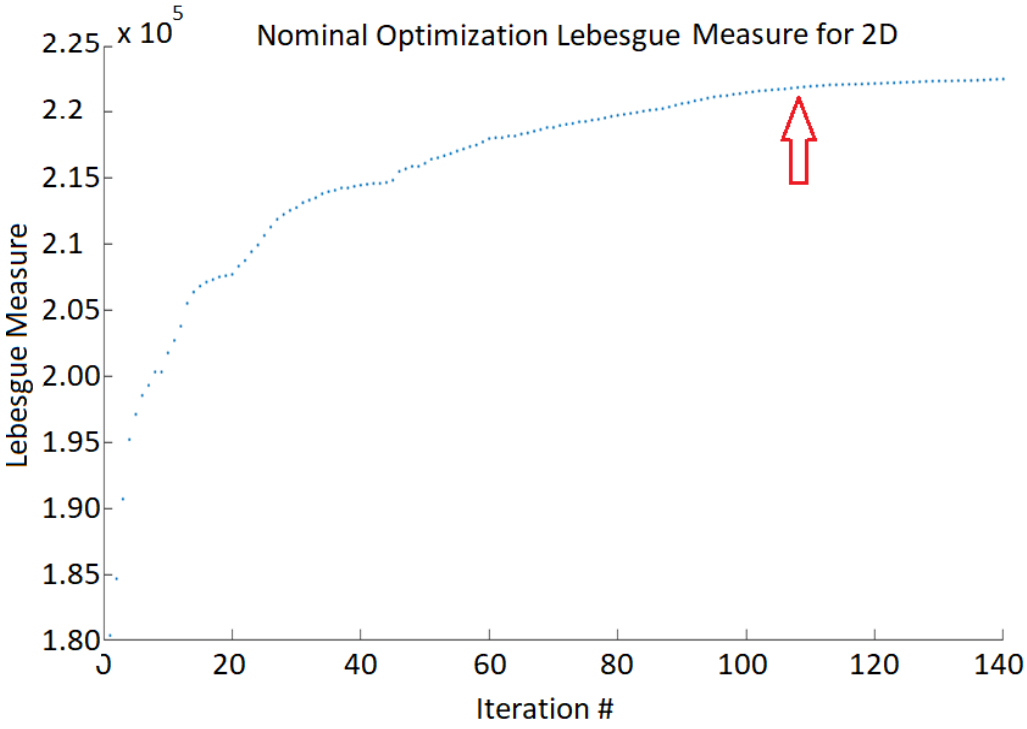

Figure 6.15. LM through the Iterations for Two-dimensional Optimization

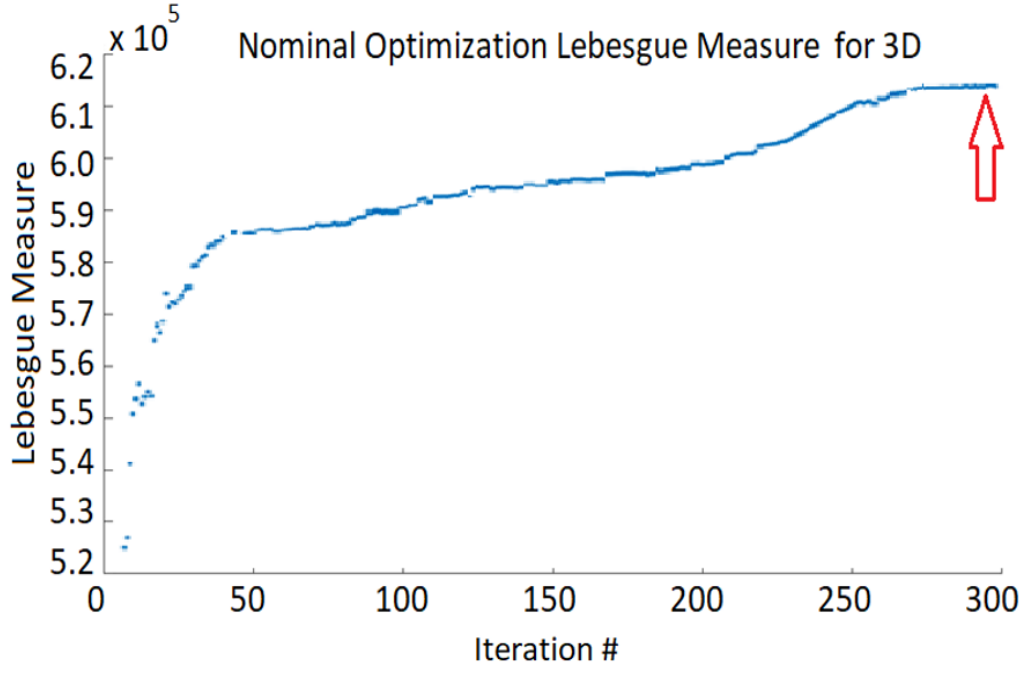

Figure 6.16. LM through the Iterations for Three-dimensional Optimization

The yield-aware optimization phase runs until two stopping criteria are satisfied. The first one considers the number of solutions in the Pareto Front that are replaced by robust ones. As discussed in Section 2, after the nominal optimization, the yield-aware optimization starts, where the yield-aware Pareto Front is generated by replacing the non-robust solutions with the robust ones. Typically, after 20 to 22 iterations (slightly varying for different methodologies and also different runs of the same methodology), the number of solutions on the robust Pareto Front can achieve 95% of the initially set population sizes, meaning the majority of the solutions have been replaced with robust ones. This percentage has been set as the first stopping criterion.

The second stopping criterion is based on the improvement of the nominal design points. The transition coefficient technique increases the probability of the replacement of the solutions by robust ones; on the other hand, it also allows the replacement of higher performance solutions with solutions having slightly worse performance. During the yield-aware optimization, the Pareto Front not only improves in terms of the number of robust solutions, but also keeps improving the nominal performances. The LM metric is evaluated after each iteration to check if the starting point value (LM obtained after the last iteration of the nominal optimization) has been reached. Typically, between 23 and 26 yield-aware iterations, this second stopping criterion is also met. Once both criteria are satisfied, the yield-aware optimization phase is stopped to output the final yield-aware PF.

6.3.2.3. Determination of the Transition Coefficient

As Table 6.19 suggests, after applying the stopping criterion which corresponds to 105 nominal iterations, the selection of the transition coefficients as 0.96/1.04 is the optimal case. Using these values, not only 96 different robust solutions on the PF could be achieved (thanks to high replacement probability), but also a quite high LM ($2.215 \, . \, 10^5$).

6.3.2.4. QMC Sample Size Selection

An ideal way of selecting the QMC sample size is decreasing the sample size as much as possible, but also guaranteeing that the sampling leads to a good approximation of the probability distribution function. For that purpose, several QMC sample sizes have been

compared by checking the variability information reflected on the different objectives using the QQ-Plot technique as in Chapter 5. One example for the QQ-Plot for different QMC sample sizes is given in Figure 6.17, which shows the variability behaviour of the total measurement range of the MEMS accelerometer system. The horizontal axis shows the standard normal quantiles, known as sigma or standard deviation in normal distribution, while the vertical axis is the original simulated values for one design point with process variations. The red line shows the quantiles of an ideal normal distribution function.

QQ-Plots have been obtained for QMC sample sizes of 25, 35, 50, 75, 100 and 1000. An acceptable QMC sample size means a good coverage of the variation space; hence, an accurate yield estimation for the related objective values. As it can be seen from Figure 6.17, decreasing the QMC sample size to below 35 drastically decreases the variation space covered. Increasing the QMC sample size increases the variation space covered; however, a very large sample size will require very high number of electrical simulations worsening the efficiency of the optimization drastically. The overall robust optimization time required for a two-dimensional, 100 population size Pareto Front is 5 hours 22 minutes for a QMC sample size of 50. Increasing the QMC sample size to 100 increases the overall robust optimization time to 9 hours 55 minutes with only a marginal improvement in the yield estimation, as the variation spaces shown in the Figure 6.17 are very similar. A QMC sample size of 1000 on the other hand, will enhance the accuracy of the yield estimation however, an estimated robust optimization time is slightly above 4 days, making it not a practical solution. It should be noted that the optimizations are run on a 1.9 GHz i3-3227 microprocessor.

In order to have a reasonable accuracy for yield estimation and also practical optimization times to obtain yield-aware Pareto Fronts, the QMC sample size has been selected as 50 for two-dimensional optimization. For three-dimensional optimization, on the other hand, a technique based on a linear increment of the QMC sample size has been applied at the yield-aware optimization iterations. Using this method, the QMC sample size per solution starts with 46 and goes up to 70 at the last iteration, with a linear increment of one sample per generation. With this methodology, replacements of the robust solutions are realized at the first iterations while the last iterations with higher QMC sample size guarantee the accuracy of the yield estimation. This methodology has been proposed as a generic technique and implemented on three-dimensional yield-aware optimization of the MEMS accelerometer system.

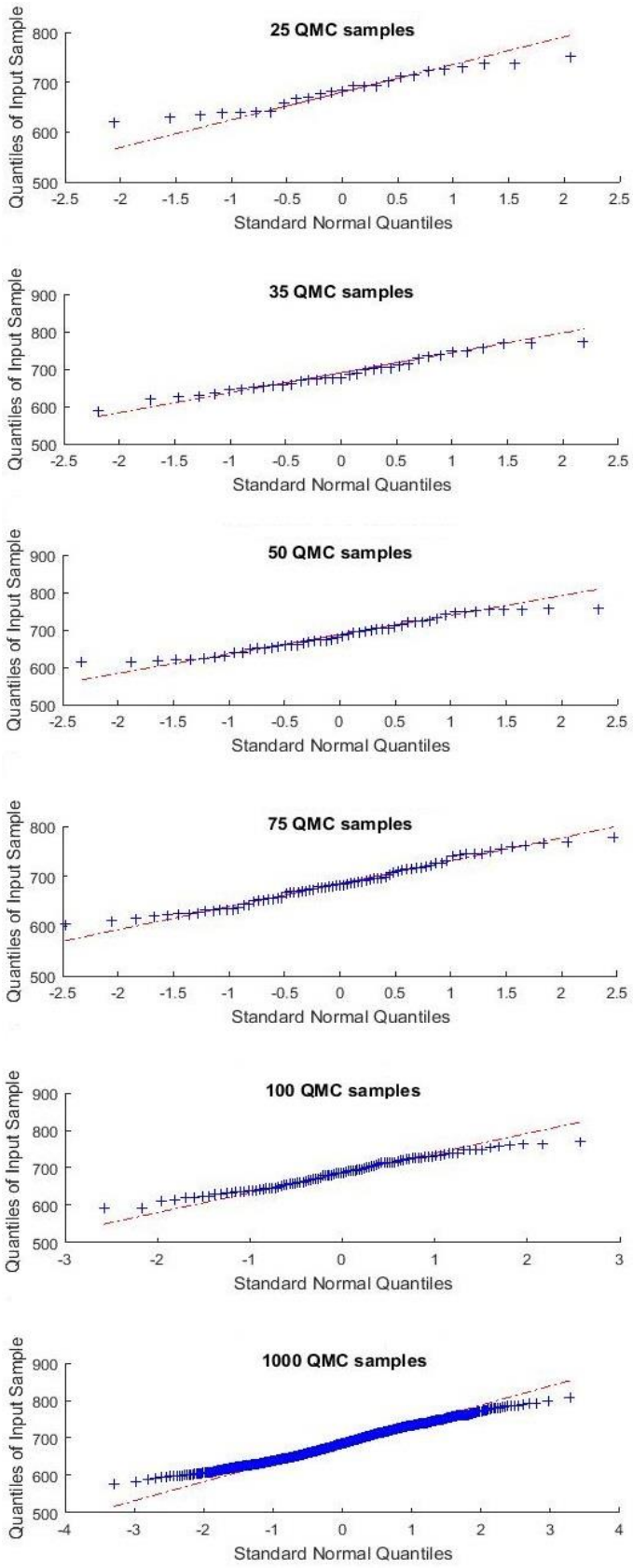

Figure 6.17. QQ-Plots obtained by Different QMC Sample Sizes for the "Measurement Range" Objective of the MEMS Accelerometer System

6.3.2.5. Selection of the Yield Optimization Constraints for CW and IBY

For two-dimensional optimization, the CW constraint has been selected as 0.2 (meaning 20% normalized total distance) and 0.27 (meaning 27% normalized distance) for three-dimensional robust optimization of the MEMS accelerometer system. The main motivation for the selection of these values is the initial variability information of the design points obtained after nominal optimization. The average CW for the all design points on the PF has been calculated as around 0.3 for two-dimensional optimization and around 0.4 for the three-dimensional optimization. Setting up the above constraint values, the variability of the solutions generated after yield-aware optimization is guaranteed to have an improvement of above 30%, compared to the results of the nominal optimization.

For the selection of the IBY on the other hand, a practical value of 90% yield has been used as the constraint to decide whether a penalization will be applied to the solutions (by increasing the fitness values of the minimization problem) on the Pareto front or not. The acceptance region is defined by allowing 5% variation (higher or lower depending on the type of the objective) on the nominal solution. With the same motivation as the selection of CW constraint, the selection of these values is based on the initial variability information of the design points obtained after nominal optimization. The average yield for the all design points on the PF has been calculated as around 65% for two-dimensional optimization and around 66% for the three-dimensional optimization. Setting up the above yield constraint for each single solution on the PF, the yield of all the solutions generated after yield-aware optimization is guaranteed to be above 90% for the acceptance region defined.

6.3.3. Selection of the Design Variables and Objectives

The optimization loop includes 20 different design variables. Seven of them (beam and mass thickness, width and length values, as well as the capacitive gap) correspond to the MEMS sensor. These 7 design variables are used in order to obtain the performances such as noise and area using an analytical model, as well as the capacitance values which are used to generate the capacitive interface of the MEMS sensor. This interface circuit is integrated into the netlist of the read-out circuitry in order to simulate the system level performances, in the same optimization loop. The amplifier topology used is given in Figure 2.5. The 13

electrical design variables are the bias current, the feedback capacitance, the feedback resistance, transistor width and length values. There are 13 different transistors in the amplifier. 5 different W and 5 different L values have been defined as design variables since the transistors are in general working as pairs with the same geometry to create the current mirrors. All the 13 transistors use a certain factor of W and L values that are optimized as design variables. Using these 13 design variables and the capacitive interface generated by the sensor evaluations, the system level performance is evaluated. Table 6.20 shows the system design variables and their allowable ranges. The optimization for both nominal and yield-aware phase is realized by using these 20 design variables.

Table 6.20 Design variables and their allowable ranges

Design Variable	**Minimum Value**	**Maximum Value**
$L_{mass}, W_{mass}, W_{beam}$	200 µm	3 mm
t_{mass}	250 µm	500 µm
L_{beam}	200 µm	2 mm
t_{beam}	50 µm	150 µm
C_{gap}	0.5 µm	1.5 µm
$W_{transistors}$	0.5 µm	300 µm
$L_{transistors}$	0.35 µm	30 µm
R_F	10 kΩ	1 MΩ
C_F	0.1 pF	10 pF
I_{bias}	0.5 µA	2.5 mA

In order to embed the variation information for the MEMS sensor, 3-sigma process variation data from the Tubitak cleanroom facilities has been used. The process variations considered for the mass and beam structures are the geometrical variations which take both lithography and etch variations into account. For the thickness of the mass and the beam structures, the variations of the deep silicon etch process has been used. The variations on the capacitive gap are defined by the deposition variations of the sacrificial oxide layer which is later removed in order to complete the formation of the capacitive gaps between the electrodes. For the electrical simulations, the technology used is a 0.35 µm CMOS technology. The variation library for this 0.35 µm technology, which includes several process parameters for Monte Carlo simulations, has also been used in the optimization loop,

to evaluate the variabilities in the system performance. This applies to both the transistors and the passive structures. It should be noted that the variations of all transistors in the amplifier have been used, in order to calculate the yield with full accuracy. Table 6.21 shows the variation information of the design variables and the libraries they were obtained from.

Table 6.21. Design variables and their variation information

Design Variable	**Variation type**	**Library**
$L_{mass}, L_{beam}, W_{mass}, W_{beam}$	Geometry: Litho + Etch	YITAL Lab.
t_{mass} , t_{beam}	Si etch variation	YITAL Lab.
C_{gap}	SiO2 deposition variation	YITAL Lab.
$W_{transistors}, L_{transistors}$	0.35 µm technology variations	0.35µm CMOS
R_F , C_F	0.35 µm technology variations	0.35 µm CMOS

Several MEMS accelerometer system performances are obtained after the electrical simulations and model evaluations. They are the overall system manufacturing cost, total system noise, phase shift delta, measurement range and the power consumption. The total system cost, on the other hand, is calculated as given in Equation (6.6). In two-dimensional robust optimization, the total system noise and the measurement range have been selected as the objectives for which the yield-aware PFs are synthesized for. For three-dimensional robust optimization, phase shift delta has been added as the third objective while all other specifications are defined as system constraints. Table 6.22 shows the system level specifications for two-dimensional robust optimization while Table 6.23 shows the system level specifications for three-dimensional robust optimization of the MEMS accelerometer system.

Table 6.22. System level specifications for two-dimensional optimization

Performance - System	**Value**
Total System Noise ($\mu g/\sqrt{Hz}$)	Minimize (Objective 1)
Measurement Range (g)	Minimize (Objective 2)
Phase Shift Delta (º)	< 3
Power Consumption (µW)	< 500
System Manufacturing Cost (USD)	< 5

Table 6.23. System level specifications for three-dimensional optimization

Performance - System	Value
Total System Noise ($\mu g/\sqrt{Hz}$)	Minimize (Objective 1)
Measurement Range (g)	Minimize (Objective 2)
Phase Shift Delta (º)	Minimize (Objective 3)
Power Consumption (µW)	< 500
System Manufacturing Cost (USD)	< 5

6.3.4. Optimization Results for Two-Dimensional Optimization of the Capacitive MEMS Accelerometer System

Two-dimensional robust optimization of the MEMS accelerometer system has been run using a population size of 100. Initially, the nominal optimization is run until the LM-based stopping criterion is satisfied. This is based on calculating the LM value at each iteration and checking if the improvement compared to the average of past 20 iterations is below 0.15%. After the nominal optimization phase, the transition coefficients are applied and yield-aware iterations, including the variability information of the CMOS, passive devices and the MEMS sensor are run, in order to obtain robust MEMS design points on the final PF. Both CW and IBY methodologies have been implemented and compared for yield optimization. The QMC sample size used for the two-dimensional yield-aware optimization is 50. The yield-aware optimization results have also been compared with the results obtained with nominal optimization to show the improvements on the device robustness. The acceptance region for yield calculations is defined by allowing 5% objective variation based on the nominal design points obtained. For the nominal optimization results which are compared with CW and IBY in Tables 6.24 to 6.28, some extra nominal optimization iterations have been run (equal to the number of yield-aware iterations determined by the two different stopping criteria) after the stopping criterion of nominal optimization is satisfied, in order to make the overall number of iterations, hence the number of simulations for the nominal design points, equal. In order to have a fair comparison, 8 different design points obtained using different techniques, which have similar system noise and measurement range values have been identified and shown in Table 6.24. The Figure-of-Merit for two-dimensional optimization is $Measurement\ Range/System\ Noise$.

Table 6.24. Comparison of eight adjacent solutions obtained with two different techniques

Design Point #1	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	20.17	20.25	20.11
Measurement Range (g)	170.0	169.6	170.1
FOM	8.43	8.38	8.46
Yield	81.88%	90.12%	69.14%
Design Point #2	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	29.73	29.82	29.68
Measurement Range (g)	285.8	284.8	285.8
FOM	9.61	9.55	9.63
Yield	83.44%	90.72%	63.12%
Design Point #3	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	39.98	40.01	39.94
Measurement Range (g)	407.0	404.9	408.2
FOM	10.18	10.12	10.22
Yield	82.65%	90.81%	66.24%
Design Point #4	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	50.54	50.87	50.52
Measurement Range (g)	535.6	534.8	536.9
FOM	10.60	10.51	10.63
Yield	80.77%	90.44%	58.45%
Design Point #5	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	60	59.98	59.96
Measurement Range (g)	649.4	643.6	650.6
FOM	10.82	10.73	10.85
Yield	80.61%	91.83%	57.45%
Design Point #6	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	70.44	70.55	70.46
Measurement Range (g)	769.2	766.9	772.2
FOM	10.92	10.87	10.96
Yield	80.41%	92.22%	62.71%
Design Point #7	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	80.22	80.46	80.11
Measurement Range (g)	886.8	883.7	889.6
FOM	11.05	10.98	11.10
Yield	81.44%	90.20%	66.34%
Design Point #8	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	90.45	90.48	90.39
Measurement Range (g)	990.6	985.3	991.6
FOM	10.95	10.89	10.97
Yield	82.65%	94.89%	68.84%

Table 6.25. Comparison of the average yield and variation values for all the solutions on the two-dimensional PFs

Performance \ Technique	CW	IBY	Nominal
2D Variation: Cloud Width (CW)	19.71%	17.50%	30.17%
Yield	81.23%	90.55%	65.10%

The overall yield-aware optimization time required has an average of 5 hours 25 minutes using a 1.9 GHz i3-3227 microprocessor.

The results suggest that for both CW and IBY techniques, the nominal design points generated by the yield-aware optimization tool are comparable with the outputs of a nominal optimization but with a much better yield performance. Hence, the robustness of the solutions on the final PF has been dramatically improved. Using the CW technique, the yield obtained is slightly worse than the PF obtained by the IBY; however, the nominal design points are slighty better in terms of performance.

6.3.5. Optimization Results for Three-Dimensional Optimization of the Capacitive MEMS Accelerometer System

Three-dimensional robust optimization of the MEMS accelerometer system has been performed by considering the phase shift delta as the third objective and using a population size of 200. Very high population sizes will require a high number of simulations, increasing the overall optimization time. Especially, considering the yield-aware QMC simulations, the population size should not be kept too high. A small population size, on the other hand, will not be able to capture the entire search space; hence, lowering the diversity of the solutions on the final PF.

Identical to the two-dimensional optimization, initially, a nominal optimization is run until the stopping criterion is satisfied. After applying the transition coefficients, 25 yield-aware iterations are realized to obtain a three-dimensional PF with robust design points. CW and IBY methodologies have been both adopted in the optimization loop and compared for yield optimization. The QMC sample size used for the yield-aware

optimization starts with 46 points and increases up to 70 points at the last iteration, with a linear increment of one sample per iteration. The goal is guaranteeing the accuracy of the yield values calculated at the final iterations while the first iterations can realize the replacements of the robust solutions relatively faster. This methodology has been applied to three-dimensional optimization only in order to enable a higher QMC sample size, which is required due to increased number of dimensions, with a reasonable optimization time. The overall yield-aware optimization time required has an average of 15 hours 20 minutes using a 1.9 GHz i3-3227 microprocessor.

The nominal optimization results have also been used for the yield comparisons. For the nominal optimization results obtained, some extra iterations (equal to the number of yield-aware iterations for robust optimization) have been run after the stopping criterion is satisfied, just like in the two-dimensional PF comparisons.

A direct comparison of 3 different design points per each yield optimization technique, with a very similar system noise, measurement range, and phase shift delta performance has been obtained as given in Table 6.26. Table 6.27, on the other hand, shows the average yield values and the variations, defined as CW, obtained from all the solutions on the final PF.

The Figure-of-Merit for three-dimensional optimization is calculated as:

$$FOM_{3D} = Measurement\ Range/(Total\ System\ Noise * Phase\ Shift\ Delta) \quad (6.8)$$

The acceptance region for yield calculations is again defined by allowing 5% objective variation based on the nominal design points obtained.

The results for three-dimensional robust optimization also suggest that for both CW and IBY techniques, the nominal design points generated by the yield-aware optimization tool are comparable with the outputs of a nominal optimization but with a dramatically better yield performance. The robustness of the solutions on the final PF has been improved with both methodologies. The FOMs obtained with the CW technique are slightly better than the ones obtained with the IBY; however, the yield of the solutions are slightly worse. Hence,

the same trade-off observed in two-dimensional optimization is also valid for three-dimensional optimization.

Table 6.26. Comparison of three adjacent solutions obtained with two different techniques

Comparison #1	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	18.08	17.77	19.08
Measurement Range (g)	103.40	96.40	97.80
Phase Shift Delta (º)	1.86	1.81	1.62
FOM	3.07	3.00	3.16
Yield	80.00%	91.42%	65.72%
Comparison #2	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	58.87	59.33	58.31
Measurement Range (g)	450.60	450.10	457.40
Phase Shift Delta (º)	0.58	0.58	0.59
FOM	13.20	13.08	13.30
Yield	82.85%	90.00%	67.14%
Comparison #3	**CW**	**IBY**	**Nominal**
Noise ($\mu g/\sqrt{Hz}$)	91.57	94.32	97.85
Measurement Range (g)	912.40	922.12	903.60
Phase Shift Delta (º)	1.44	1.43	1.32
FOM	6.92	6.84	7.00
Yield	81.42%	91.42%	62.86%

Table 6.27. Comparison of the average yield and variation values for all solutions on the three-dimensional PFs

Performance \ Technique	**CW**	**IBY**	**Nominal**
3D Variation: Cloud Width (CW)	25.94%	22.02%	40.58%
Yield	80.48%	90.69%	66.65%

Figures 6.18, 6.19, and 6.20 show the variations on the system performance for the first design points in Table 6.26 obtained using the CW methodology and the nominal optimization.

The QMC sample size is 70 as the results are extracted from the final PF. The variations on the nominal performances are shown as two-dimensional projections of the three objectives.

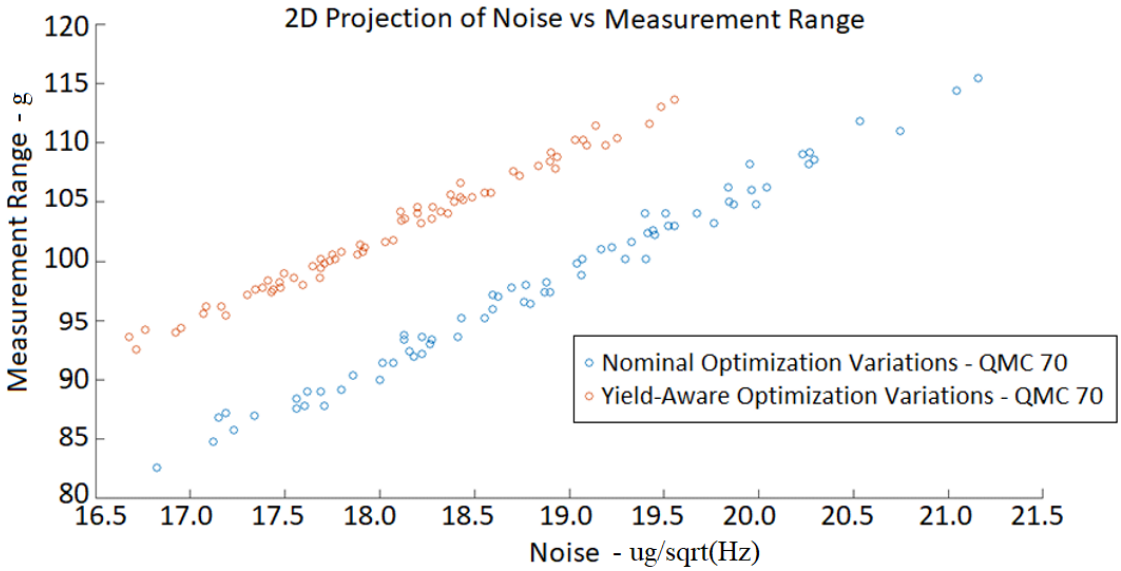

Figure 6.18. Noise vs Measurement Range Performance Variation of a Single Solution for CW vs Nominal Optimization

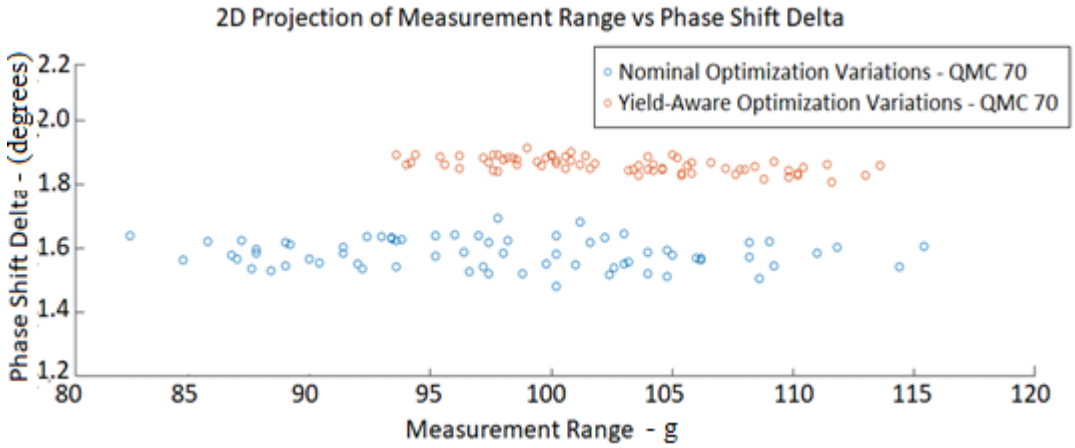

Figure 6.19. Measurement Range vs Phase Shift Delta Performance Variation of a Single Solution for CW vs Nominal Optimization

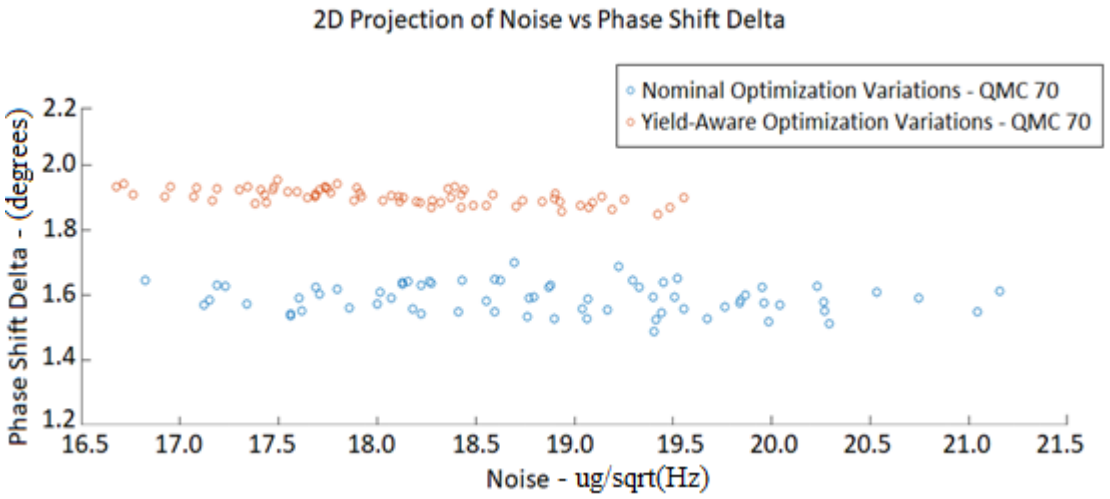

Figure 6.20. Noise vs Phase Shift Delta Performance Variation of a Single Solution for CW vs Nominal Optimization

The results demonstrate the decrease in the variability obtained by using the yield-aware optimization methodology proposed.

In order to guarantee the accuracy of the yield estimation, a 10,000 sample MC simulation has been run for the first design points compared in Table 6.26. Table 6.28 shows the yield values obtained using 10,000 sample MC analysis.

Since the QMC is deterministic and has no natural variance, there is no practical way to determine the confidence interval of the estimated yield. This issue has been addressed in [62] and [63] and a solution has been proposed and implemented by running a scrambled QMC, in which samples are randomly permuted several times to obtain a stochastic variance. In order to obtain the confidence intervals for the estimated yield, this technique has been applied for 20 different scrambled QMC on the nominal design parameters of each design point. The confidence intervals of the yield estimations with 99% confidence level are also given in Table 6.28.

The confidence interval for MC (10.000 samples) with 99% confidence level, on the other hand, is calculated based on [50] and included in Table 6.28.

Table 6.28. Comparison of the yield obtained using QMC and MC for the first solutions obtained with different techniques

Comparison #1	**CW**	**IBY**	**Nominal**
Noise	18.08	17.77	19.08
Measurement Range	103.40	96.40	97.80
Phase Shift Delta	1.86	1.81	1.62
FOM	3.07	3.00	3.16
Yield – QMC 70 Samples	80.00%	91.42%	65.72%
Yield – MC 10,000 Samples	76.15%	88.72%	61.63%
Confidence Interval (99%) for 70 sample QMC (20 runs)	74.61%≤x ≤80.25%	86.92%≤x≤91.50%	60.61%≤x≤66.07%
Confidence Interval (99%) for 10,000 sample MC	75.06%≤x ≤77.24%	87.91%≤x≤89.53%	60.41%≤x≤62.85%

The results show that the yield calculations based on 70 sample QMC and 10,000 sample MC (which can be considered as exact yield) are quite similar to each other, with around and less than 4% deviation on the yield estimation. Hence, the QMC-based methodology can give a reasonably accurate estimation of the yield for the MEMS accelerometer system optimized. Moreover, all MC results are in the confidence interval calculated.

The reported yield values in Table 6.28 should not be interpreted as a target yield in a practical application. The goal of the yield-aware optimization is to get a good balance between the maximization/minimization of the objectives and the minimization of their variabilities. Once a design point is obtained, its real yield value will also depend on the definition of the acceptance region. Therefore, the yield estimations of Table 6.28 should not be interpreted as a target yield value but as a demonstration that for an equivalent acceptance region definition, the proposed techniques dramatically increase the yield value. For a different acceptance region definition, the yield values will differ. And the opposite, for a given yield target, the acceptance region, and hence, the achievable specifications can be determined.

7. CONCLUSIONS

The complex trade-off exploration in system design for MEMS, which composes a mechanical sensor or actuator with an electronic circuit, requires a lot of expert knowledge, which implies long design times and increased cost especially due to the physical heterogeneity involved. Industry has been traditionally following ad-hoc approaches based on the separate design of the MEMS sensor and the circuitry by different design teams and then, combining these two parts. However, combining these separately designed blocks has several issues. A first problem is that inappropriate partitioning of the system specifications among the mechanical and electronic parts might lead to non-optimal system performance. Another problem, that can appear during the composition of independently designed blocks, is the risk of violating some system level constraints. Hence, design methodologies that simultaneously considers both the mechanical and the electronic parts in order to develop a multi-domain MEMS synthesizer is needed for system level optimization of MEMS.

The same story for the lack of optimization tools does not apply to the analog circuit optimization which already had quite a lot of interest and effort for the implementation of the optimization tools. On the other hand, there is an obviously challenging problem and a hot topic for the analog circuit optimization, which is the robustness of the designs, hence the yield optimization.

Based on the two motivations discussed above, this dissertation has mainly focused on implementing generic methodologies which can be used for yield-aware optimization of analog circuits and MEMS.

Initially, the evaluation of the multi-objective optimization tool MOEA/D has been performed by running several high-dimensional (up to five dimensions) optimizations. Pareto Fronts with different dimensions have been compared (using several performance comparison metrics: IGD, Coverage Set, Schott's Spacing) to investigate the performance degradation with respect to increased number of dimensions. The results show that increased number of dimensions for the optimization of the folded-cascode amplifier do not significantly deteriorate the quality of the Pareto Front (e.g., 27% vs. 28% Coverage Set as

an average of 3 runs for three- and four-dimensional optimizations, respectively) ; hence MOEA/D has been qualified as a powerful optimizer for the synthesis of the complex circuits and systems, such as yield-aware optimization of the analog circuits and MEMS.

In the thesis, multi-objective optimization of a MEMS system by jointly optimizing the mechanical sensor and the analog read-out circuitry has been performed for the first time in the literature. The design points obtained using the proposed joint optimization have been compared with the design points obtained by using the conventional design methodology where the sensor and circuit are optimized separately and then combined. The results show very good improvement in the system performance (e.g. up to 42% improvement in manufacturing cost for the same noise constraint: 24 cents per chip instead of 41 cents)

In the thesis, several yield-aware optimization techniques have been implemented and compared. A novel yield constraint technique called cloud width (CW), which is based on the span of the performance variability, has been developed and compared with two different methodologies (IBY and WCPF) which already exist. These techniques have been compared by running yield-aware optimization of an analog circuit (folded-cascode amplifier) and also MEMS (sensor and read-out circuitry). Comparisons using the analog circuit have been performed between these three techniques and the emulation of a designer's approach. The design points obtained using the three yield-aware optimization methodologies have also been compared with the design points obtained with nominal optimization only. The implemented yield-aware optimization methodologies show a dramatic improvement on the robustness (e.g., 10.5% normalized performance variation on a design point instead of 27.5% performance variation) of the designs. All three techniques show very good results, and, in many cases, the novel CW performs the best or comparable to other techniques.

Similar comparisons have been performed by running yield-aware optimization of a MEMS accelerometer (also first time in the literature) as well. Design points obtained with IBY and CW techniques have been compared with the design points obtained running nominal optimization only and a very good improvement on yield (above 90% yield instead of 65%) has been achieved.

In order to integrate the yield-aware optimization methodologies into the optimization loop and also to develop a tool which is as generic as possible, several techniques such as the “transmission factor” to enhance the replacement mechanism during yield-aware optimization phase, QMC sample size selection for the statistical analysis of the analog circuits and MEMS, Lebesgue Measure based stopping criterion etc. have been developed. The results showing the improvements on the diversity and robustness of the final Pareto Front has been reported in the thesis.

There are several works that can be carried out to improve this study. RF circuits, for instance, require a lot of expert knowledge and design optimization tools which can perform the trade-off exploration are needed. As demonstrated in Chapter 3, the RF circuits are also highly impacted by the process variations which makes yield-aware optimization tools essential for robust RF circuit synthesis. As a future work, the implemented methodology can be used to test the performance on RF circuits.

Another study that can be considered as a future work is the integration of the post-layout simulations into the yield-aware optimization loop. The use of the extracted parasitics will enhance the accuracy of the final design; however, considering the high number of simulations that is run in the optimization, it is not practical to implement a fully post-layout simulation based methodology. The best alternative would be running the last few iterations of the yield-aware optimization phase with post-layout simulations. Doing that, the accuracy of the circuit performances and the yield estimation will be enhanced.

Finally, thanks to the novel methodologies proposed and implemented, during the thesis study, three journal and three conference papers have been published. The publication list is also given in the dissertation.

REFERENCES

1. Toumazou, C., Makris, C.A., " Analog IC Design Automation: Part I - Automated Circuit Generation: New Concepts and Methods", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 2, February 1995

2. Plas, V. D. , Gielen, G. , Sansen, W. , *A Computer – Aided Design and Synthesis Environment for Analog Integrated Circuits*, Kluwer Academic Publishers, 2002.

3. Analog Devices, *Data Sheet for the ADXL355 3 - Axis MEMS Accelerometer*, https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_355.pdf, accessed on 14 November 2019

4. Pak, M., Fernandez, F. V., Dundar, G., "Yield-aware Multi-Objective Optimization of a MEMS Accelerometer System Using QMC - Based Methodologies", *Microelectronics Journal*, vol. 103, September 2020

5. Pak, M., *An Enhanced Multi - Objective Evolutionary Algorithm MOEAD-DE for the Applications of Analog Sizing with both W/L and Operating Point Driven (OPD) based Methods*, M.Sc. Thesis, Bogazici University, 2010

6. Pak, M., Fernandez, F. V., Dundar, G., " A Novel Design Methodology for the Mixed-Domain Optimization of a MEMS Accelerometer", *Integration, the VLSI Journal*, vol. 62, pp. 314-321, April 2018

7. Tan, T., Roy, S., Thuy, N., Huynh, H.T., "Streamlining the design of MEMS devices: An Acceleration Sensor", *IEEE Circuits and Systems Magazine*, vol. 8, pp. 18–27, 2008

8. Zhang, Y., Kamalian, R., Agogino, A.M., Sequin, C. H., "Hierarchical MEMS Synthesis and Optimization", *Proceedings of SPIE* - International Society for Optical Engineering, vol. 5763, May 2005

9. Neul, R., et al., “A Modeling Approach to Include Mechanical Microsystem Components into the System Simulation”, *Proceedings of Design, Automation and Test in Europe Conference, DATE*, pp. 510-517, 1998.

10. Pak, M., Fernandez, F. V., Dundar, G., “Optimization of a MEMS accelerometer using MOEA/D Evolutionary Algorithm ”, *14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, SMACD*, pp. 1-4, 2017

11. Fan, Z., et al., “ Hierarchical Evolutionary Synthesis of MEMS”, *IEEE Congress on Evolutionary Computation*, pp. 2320–2327, 2004

12. Mukherjee, T., Fedder, G., “Hierarchical Mixed-Domain Circuit Simulation, Synthesis and Extraction Methodology for MEMS”, *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 21, pp. 233-249, 1999

13. Zhao, C., Kazmierski, T. J., “ Automated Performance Optimization and Layout Synthesis of MEMS Accelerometer with Sigma-Delta Force – Feedback Control Loop’, *Proceedings of IEEE Behavioral Modeling and Simulation Workshop*, pp. 19-24, 2008.

14. Klaus, J., Paris, R., Sommer, R., “ Systematic MEMS ASIC Design Flow using the Example of an Acceleration Sensor ”, *13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1-4, 2016

15. Martens , E. , Gielen, G., “ Classification of Analog Synthesis Tools based on their Architecture Selection Mechanisms ”, *Integration – the VLSI Journal* 41, vol. 41, pp. 238-252, 2008

16. Pak, M., Yarimbiyik, A. E., Dundar, G., Fernandez, F. V., “Statistical analysis of active and passive RF devices”, *Proceedings of International Conference on Advanced Semiconductor Devices & Microsystems (ASDAM)*, 2014, pp.1–4

17. Pak, M., Fernandez, F. V., Dundar, G., "Comparison of QMC-based Yield- aware Pareto Front Techniques for Multi-objective Robust Analog Synthesis", *Integration , the VLSI Journal* , vol. 55, no. 9, pp.357-365, September 2016

18. Burcea, F., Herrman, A., Li, B., Graeb, H., " MEMS – IC Optimization Considering Design Parameters and Manufacturing Variation from both Mechanical and Electrical Side ", *25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 625-628, 2018

19. Zhang, Q., Li, H., " MOEA/D : A Multiobjective Evolutionary Algorithm based on Decomposition', *Proceedings of IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007

20. Xue, F., *Multi-Objective Differential Evolution: Theory and Applications*, Ph.D. Thesis, Rensselaer Polytechnic Institute, September 2004

21. Pohlheim, H., " Introduction Evolutionary Algorithms : Overview, Methods and Operators " , *Documentation for Genetic Algorithms Toolbox for Matlab* , http://www.geatbx.com/docu/algindex.html , accessed on 2 September 2020

22. Branke, J., Deb, K., Dierolf, H., Osswald, M., *Finding Knees in Multi - objective Optimization,* KanGAL Report No: 2004010

23. Schaffer, J. , " Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Proceedings of the First Int. Conference on Genetic Algorithms*, pp. 93 - 100, 1985

24. Murata, T., Ishibuchi, H., " MOGA: Multi-objective Genetic Algorithms," *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, pp. 289-. 297, 1995

25. Srinivas, N., Deb, K., " Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, vol. 2, pp. 221-248, 1995.

26. Deb, K., Prapat, A., Agarwal, S., Meyarivan,T., " A Fast and Elitist Multiobjective Genetic Algorithm: NSGAII", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002

27. Zitzler, E., Laumanns, M., Thiele, L., " SPEA2 : Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization " , *Proceedings of the EUROGEN*, 2001

28. Liu, B., Fernandez, F. V., Zhang, A., Pak, M., Sipahi, S., Gielen, G., "An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing", *IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-7, 2010

29. Pak, M., Fernandez, F. V., Dundar, G., "Integration of QMC Based Yield-Aware Pareto Front Techniques on MOEA/D for Robust Analog Synthesis", *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 1-4, 2015

30. Asafuddoula, M., Singh H. K., Ray, T., "Six - Sigma Robust Design Optimization using a Many - objective Decomposition Based Evolutionary Algorithm", *IEEE Transactions on Evolutionary Computation*, vol. 19, pp. 490-507, 2014

31. Zitzler, E., Thiele, L., " Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach ", *IEEE Transactions of Evolutionary Computation*, vol. 3, pp. 257–271, 1999.

32. CEC 2009, *Benchmark Problems and Test Conditions*, CEC 09 MOEA Competition, http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm, accessed on 14 November 2019

33. Knowles, J., Corne, D., "On Metrics for Comparing Nondominated Sets," *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, vol. 1, pp. 711-716, 2002

34. Micheal, M., *Modelling Statistical Variability Within Circuits Using Nano - CMOS*

Technologies, Ph.D. Thesis, University of Southampton, June 2012

35. Beeresha, R. S., Khan A. M., Reddy, H. V. M., "Design and EM - Simulation of MIM Capacitor," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 1644-1649, 2017

36. Agarwal, A., et al., "Path - based statistical timing analysis considering inter- and intra-die correlations ", *ACM / IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 16-21, 2002

37. Doh, J. S., Kim, D. W., " A Unified Statistical Model for Inter - Die and Intra - Die Process Variation", *2005 International Conference on Simulation of Semiconductor Processes and Devices*, Tokyo, Japan, 2005, pp. 131-134

38. Singhee, A., Rutenbar, R. A., " Why Quasi-Monte Carlo is Better than Monte Carlo or Latin Hypercube Sampling for Statistical Circuit Analysis", *IEEE Transactions on CAD* , vol. 29, no.11, 2010

39. McKay, M. D., Beckman, R. J., Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code", *Technometrics American Statistical Association*, vol. 21, pp. 239–245, 1979

40. Singhee, A., Rutenbar, R. A., " From finance to flip flops: A study of fast quasi Monte Carlo methods from computational finance applied to statistical circuit analysis," *IEEE International Symposium on Quality Electronic Design*, 2007, pp. 685-692

41. Li, X., Le, J., Pileggi, L. T. , *Statistical Performance Modeling and Optimization*, Foundations and Trends in Electronics Design Automation, Now Publishers, 2007

42. Okobiah, O., Mohanty, S., Kougianos, E., "Fast Statistical Process Variation Analysis using Universal Kriging Metamodeling: A PLL Example ", *Midwest Symposium on Circuits and Systems*, pp. 277-280, 2013

43. De Cock, D. R., *Kriging as an Alternative to Polynomial Regression in Response Surface*

Analysis, Ph.D. Thesis, Iowa State University, 2013

44. AMS, *Process Technology* , https://ams.com/process-technology, accessed on 20 November 2019

45. Synopsys, *TCAD,* https://www.synopsys.com/silicon/tcad.html,accessed on 1 December 2019

46. Sonnet, *Sonnet Software,* http://www.sonnetsoftware.com/, accessed on 1 December 2019

47. Mc Conaghy, T., Gielen, G., "Globally Reliable , Variation - Aware Sizing of Analog Integrated Circuits via Response Surface and Structural Homotopy", *Proceedings of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* pp. 1627-1640, 2009

48. Afacan, E., Berkol, G., Pusane, A. E., Dundar, G., Baskaya, F., "Adaptive Sized Quasi-Monte Carlo based Yield Aware Analog Circuit Optimization Tool", *Proceedings of European Workshop on CMOS Variability,* pp. 1-6, 2014

49. Liu, B., Fernandez, F. V., Gielen, G., "An Accurate and Efficient Yield Optimization Method for Analog Circuits based on Computing Budget Allocation and Memetic Search Technique", *Proceedings of Design Automation and Test in Europe Conference and Exhibition,* pp. 1106-1111, 2010

50. Graeb, H. E., *Analog Design Centering and Sizing,* Springer, 2007

51. Gomez, I. G., *Optimization of Analog Circuits by Applying Evolutionary Algorithms,* Ph.D. Thesis, National Institute for Astrophysics, Optics and Electronics, Mexico, 2012

52. Wilk, M. B., Gnanadesikan, R., "Probability Plotting Methods for the Analysis of Data", *Biometrika,* vol. 55, no. 1, pp. 1-17, 1968

53. Fleischer, M., *" The Measure of Pareto Optima: Applications to Multiobjective*

Metaheuristics", Technical Research Report, Army Research Laboratory, MD, 2002

54. Comsol MEMS Module Website, *Simulation of Microelectromechanical Systems,* https://www.comsol.com/mems-module, accessed 12 February 2020

55. Zhang, G., *Design and Simulation of A CMOS-MEMS Accelerometer,* Ph.D. Thesis, Carnegie Mellon University, 1998

56. Belendez, T., Neipp, C., Belendez, A. , " Large and small deflections of a cantilever beam", *European Journal of Physics,* vol. 23, no. 3, pp.371-379, May 2002

57. Strong, J., "Carbon MEMS Accelerometer", *Proceedings of the Comsol Conference,* Boston, 2011

58. Boser, B. E., Howe, R. T., "Surface Micromachined Accelerometers", *IEEE Custom Integrated Circuits Conference,* pp. 337-344, 1995

59. Acar, C. , Shkel, A. M. , " Experimental Evaluation and Comparative Analysis of Commercial Variable - Capacitance MEMS Accelerometers " , *Journal of Micromechanics and Microengineering,* vol. 13, pp. 634-645, 2003

60. Song, H., et al., " Fully Integrated Low – Noise Readout Circuit with Automatic Offset Cancellation Loop for Capacitive Microsensors", *Sensors,* vol. 15, pp. 26009-26017, 2015

61. Fang, D., *Low Noise and Low Power Interface Circuits Design for Integrated CMOS - MEMS Inertial Sensors,* Ph.D. Thesis, University of Florida, 2006

62. Afacan, E., Gonenc, B., Pusane, A. E., Dundar, G., Baskaya, F., " A Hybrid Quasi Monte - Carlo Method for Yield Aware Analog Circuit Sizing Tool " , *Design, Automation and Test in Europe Conference and Exhibition (DATE),* pp. 1225-1228, 2015

63. Morohosi, H., Fushimi, M., “A Practical Approach to the Error Estimation of Quasi - Monte Carlo Integration”, M*onte Carlo and Quasi - Monte Carlo Methods*, vol. 377 p. 390, 1998