Enhancing Variation-aware Analog Circuits Sizing

Ons Lahiouel

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

April 2017

CONCORDIA UNIVERSITY

Division of Graduate Studies

This is to certify that the thesis prepared

By: **Ons Lahiouel**

Entitled: **Enhancing Variation-aware Analog Circuits Sizing**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____________________ Dr. Tarek Zayed

____________________ Dr. Roni Khazaka

____________________ Dr. Reza Soleymani

____________________ Dr. Otmane Ait Mohamed

____________________ Dr. Eusebius J. Doedel

____________________ Dr. Sofiène Tahar

Approved by ______________________________

Chair of the ECE Department

________ 2017 ______________________________

Dean of Engineering

ABSTRACT

Enhancing Variation-aware Analog Circuits Sizing

Ons Lahiouel

Concordia University, 2017

Today's analog design and verification face significant challenges due to circuit complexity and short time-to-market windows. Moreover, variations in design parameters have an adversely impact on the correctness and performance of analog circuits. Circuit sizing consists in determining the device sizes and biasing voltages and currents such that the circuit satisfies its specifications. Traditionally, analog circuit sizing has been carried out by optimization-based methods, which of course will still be important in the future. Unfortunately, these techniques cannot guarantee an exhaustive coverage of the design search space and hence, are not able to ensure the non-existence of higher quality design solutions. The sizing problem becomes more complicated and computationally expensive under design parameters fluctuation. Indeed, existing yield analysis methods are computationally expensive and still encounter issues in problems with a high-dimensional process parameter space. In this thesis, we present new approaches for enhancing variation-aware analog circuit sizing. The circuit sizing problem is encoded using nonlinear constraints. A new algorithm using Satisfiability Modulo Theory (SMT) solving techniques exhaustively explores the analog design space and computes a continuous set of feasible sizing solutions. Next, a yield optimization stage aims to select the candidate design solution with the highest yield rate in the presence of process parameters variation. For this purpose, a novel method for the computation of parametric yield is proposed. The method combines the advantages of sparse regression and SMT solving techniques. The key idea is to characterize the failure regions as a collection of hyperrectangles in the parameters space. The yield estimation is based on a geometric calculation of probabilistic volumes subtended by the located hyperrectangles. The method can provide very large speed-up over Monte Carlo methods, when a high prediction accuracy is required. A new approach for improving analog yield optimization is also proposed. The optimization is performed in two steps. First, a global optimization phase samples the most potential optimal sub-regions of the feasible design space. The global search locates a design point near the optimal solution. Second, a local optimization phase uses the near optimal solution as a starting point. Also, it constructs linear interpolating models of the yield to explore the basin of convergence and to reach the global optimum. We illustrate the efficiency of the proposed methods on various analog circuits. The application of the yield analysis method on an integrated ring oscillator and a 6T static RAM proves that it is suitable for handling problems with tens of process parameters and can provide speedup of 5X-2000X over Monte Carlo methods. Furthermore, the application of our yield optimization methodology on the examples of a two-stage amplifier and a cascode amplifier shows that our approach can achieve higher quality in analog synthesis and unrivaled coverage of the analog design space when compared to traditional optimization techniques.

To My Husband, My Mom and My Dad.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor, Professor Sofiène Tahar, for his continual support and guidance throughout my graduate studies at Concordia University. I have learned so much from his deep insights about research and strong expertise. He is always approachable, and he showed confidence on me even in the most frustrating situations which greatly helped me to stay motivated. Moreover, I am grateful for the opportunities he had provided me, like presenting in the most prestigious conferences in design automation and coaching a Master's student. In short, I will always be indebted for the time and the effort he has spent to help me complete my PhD project.

I am very pleased that Dr. Roni Khazaka has accepted to be my external PhD thesis examiner. I am also thankful to Dr. Reza Soleymani, Dr. Glenn Cowan, Dr. Otmane Ait Mohamed and Dr. Eusebius J. Doedel for serving on my doctoral advisory committee. Their comments at various stages have been significantly useful in shaping my research project. Many thanks to the School of Graduate Studies for the scholarships that allowed me to present my research at conferences and complete my thesis writing and defence.

I am thankful to many members of the Hardware Verification Group. In particular, I would like to thank Dr. Mohamed Zaki for many technical discussions, feedback and sound advices. I am also thankful to Dr. Henda Aridhi and Muhammed Shirjeel for the fruitful collaborations on various research projects. My sincere thanks to all my good friends at the Hardware Verification Group.

This thesis would not have been possible without the support and encouragement of my husband. I am also eternally grateful to my parents for providing me with unconditional love and support all my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AC	Alternating Current
AMS	Analog and Mixed Signal
ASR	Adaptive Sparse Regression
CAD	Computer-Aided Design
CDCL	Conflict-Driven Clause Learning
CMOS	Complementary Metal Oxide Semiconductor
CPR	Clustered Polynomial Regression
DC	Direct Current
DE	Differential Evolution
EC	Evolutionary Computation
EP	Evolutionary Programming
ES	Evolution Strategies
GA	Genetic Algorithm
GP	Geometric Programming
HB	Harmonic Balance
IC	Integrated Circuit
INTLAB	Interval Laboratory
IS	Importance Sampling
KCL	Kirchhoff's Current Law
LASSO	Least Absolute Shrinkage and Selection Operator
LDS	Low Discrepancy Sequences
LHS	Latin Hypercube Sampling
MARS	Multivariate Adaptive Regression Splines

MSEOA	Memetic Single Objective Evolutionary Algorithm
MC	Monte Carlo
NMSE	Normalized Mean Square Error
NN	Neural Network
OO	Ordinal Optimization
OPD	Operating Point Driven
OPF	Optimal Power Flow
PCA	Principle Component Analysis
PDF	Probability Density Function
PR	Polynomial Regression
PM	Phase Margin
PSO	Particle Swarm Optimization
PSS	Periodical Steady State
PSWCD	Performance-Specific Worst-Case Design
QMC	Quasi Monte Carlo
RBF	Radial Basis Function
RF	Radio Frequency
RReliefF	Regressional ReliefF
SAT	Satisfiability
SA	Simulated Annealing
SDP	Semi Definite Programming
SMT	Satisfiability Modulo Theory
SoC	System-on-Chip
SPICE	Simulation Program with Integrated Circuit Emphasis
SPRT	Sequential Probability Ratio Test

SQP	Sequential Quadratic Programming
SR	Sparse Regression
SRAM	Static Random Access Memorie
SVM	Support Vector Machine
TSMC	Taiwan Semiconductor Manufacturing Company
VTC	Voltage Transfer Curve

Chapter 1

Introduction

1.1 Motivation

Over the last decade, CMOS (Complementary Metal Oxide Semiconductor) technology scaling has been a primary driver of the electronics industry [3]. This scaling trend is a natural response to the continuously increasing demand for high performance and multi function consumer electronics (smart phones, wearable devices, autonomous robots, etc.). Most electronic products rely on System-on-Chip (SoC) solutions, where one integrated circuit contains the whole system function. Modern SoC designs integrate billions of transistors and contain various interactive system components, from analog/RF circuits to digital signal processing and memory blocks [4]. Apart from generating system reference clock (e.g., a phase locked loop (PLL)), increasing the power of a signal (e.g., operational amplifier) and ensuring the correct operation of the chip (e.g., biasing circuits), the analog part is indispensable for all electronic devices [4]. For example, no matter how digital our electronic devices get, they always require analog and mixed signal (AMS) interfaces that translate signals from the physical world into the digital world of electronics. Although the analog part is a small fraction of the entire integrated circuit [5], its design is usually much more time consuming and error prone than the relatively larger digital portion. It often becomes the major bottleneck that limits system performance, product yield, and time to market.

A large analog system with many transistors and passive components is not designed as a whole, but is decomposed into sub-blocks. Each sub-block will be further decomposed down to the cell level [6]. An analog cell is a small circuit having a certain basic function, such as an amplifier, a mixer, a filter, etc. Given a set of specifications and the technology used, the design flow of an analog cell is mainly composed of topology selection, circuit sizing, layout and fabrication. The parameter-level analog design flow (i.e., circuit sizing) is the process through which the biasing and sizing of all devices (transistors, capacitors, resistors, etc.) are determined such that the circuit meets its specifications. The goal of this step is to make the parameterized circuit topology satisfy the specifications as verified by circuit simulation.

Most analog circuit sizing problems can naturally be expressed as a single- or multi-objective constrained optimization problem, where the goal is to determine the sizing solution that optimizes one or multiple performance metrics, e.g., power consumption, area, etc. Despite the tremendous growth in computer-aided design (CAD) tools for circuit synthesis and optimization, the design of analog cells is still being handcrafted using a schematic capture software (usually SPICE circuit simulator [7]), and the circuit sizes are determined manually or with little automation.

One of the main reasons for this lack of automation and complexity is the limited capabilities of the optimization techniques. Although modern numerical optimizations have been introduced to analog integrated circuit (IC) sizing, the objective optimization and constraint handling abilities of most of the existing methods are still not good enough for high-performances analog circuit sizing. The large design search space and the complexity of analog characteristics make the circuit optimization process complex. Even worse, driven by the market demands and advances in fabrication technologies, the specifications of modern analog circuits are becoming increasingly stringent and consequently result in a more complicated optimization problem. Owing to this, there is a real need for a more powerful search process able to explore a large range of design variables towards improving sizing solution.

The design task becomes more difficult with the aggressive down-scaling of silicon technology, owing to the increasing process-induced variability [8]. In fact, process variation has become a major concern for today's analog circuits, due to significantly increased circuit failures and parametric yield (i.e., the probability that the circuit meets the performances constraints) loss [9]. Indeed, the variations in device size and operating point [10] are the main factors that deviate the performance of an analog circuit from its desired property [8]. For example, nanoscale transistors exhibit more mismatches, leading to random offset errors and poor gain performance [11]. Indeed, analog IC components must be designed with sufficiently high yield in light of large-scale process variations. For these reasons, it becomes important to estimate and optimize the yield both efficiently and accurately within the analog design flow [12].

This thesis is largely motivated by the powerful and new solving techniques in modern Satisfiability (SAT) Modulo Theory (SMT) [13] solvers. These solvers check the satisfiability of first-order formulas containing operations from various theories such as real numbers and integers. They are built upon a tight integration of modern

Conflict-Driven Clause Learning (CDCL)-style SAT solving techniques with interval-based arithmetic constraint solving within an SMT framework. They are capable of handling constraints containing nonlinear functions over a very large number of variables [14], which is one of the inherent characteristics of analog circuits operation/performances models. Most importantly, they can be leveraged to exhaustively explore the search space of a constraint-satisfaction system, making them a potentially appealing choice for parameters space exploration strategies of analog circuits. Though, they should be properly employed.

In this thesis, we propose new techniques that tackle several limitations of the analog circuit sizing procedure. Our ultimate goal is to ensure an exhaustive coverage of the design space using SMT solving technique. The search strategy should relieve the sizing solution from the uncertainty inherited from optimization-based methods. Once a complete set of feasible design solutions is determined, the second objective of our research is to propose a new method that estimates the circuit robustness (i.e., parametric yield) in light of process variation. The method should keep a reasonable computational cost and guarantee a good accuracy. Our third objective is to propose a new optimization technique for yield-aware circuit sizing that efficiently selects the best design solution in terms of robustness.

1.2 State-of-the-Art

In this section, we briefly review the status of existing circuit sizing, yield estimation as well as yield optimization techniques closely related to this thesis.

1.2.1 Analog Circuit Sizing

Given a circuit schematic and a set of specifications, circuit sizing denotes the task of determining the sizes and biasing of all devices such that the circuit meets the specifications. Generally, it is an optimization engine that determines these values, while the evaluation engine assesses the circuit performances [15]. Techniques that have been employed as optimization routines for analog circuits can broadly be classified into two main categories: deterministic optimization algorithms (Newton methods, Levenberg-Marquardt method, etc.) and stochastic search algorithms (evolutionary computation algorithms [2], simulated annealing [16], etc.). The main contributions for analog design techniques are surveyed in [6].

The disadvantages of deterministic algorithms, such as the requirement of a good starting point, the high probability of getting trapped into local optima, and the conditions of continuous and differentiable objective function, limit their applicability in analog design methodologies [6]. In general, the differentiability condition is satisfied, however, a large number of simulations is needed to obtain and evaluate the gradients, which becomes the bottleneck of the circuit optimization process [17]. Besides, the optimization needs to start from a good initial point and there is no guarantee that it will reach the global optimum, particularly for non-convex optimization problems [18]. Convex optimization is another deterministic approach that uses geometric programming (GP) operating on posynomial functions [19]. However, this method too, has met challenges, primarily because extensive studies have demonstrated that posynomials fail to produce accurate models for large circuits [20].

Recent advances in polynomial optimization show that the general polynomial optimization problem can be transformed to a convex problem by Semi definite-Programming (SDP) relaxations, which makes it possible to find the global optimum of the circuit optimization [20]. Unfortunately, the problem of the SDP-based polynomial optimization method is that the polynomial approximations cannot guarantee the modeling accuracy over the whole design space.

Multiple starting point optimization algorithm has also been proposed for analog circuit optimization [17] [21]. From a set of starting points, the corresponding local optima are reached by a local optimization method. The global optimum is then selected from these local optima. If one starting point is located in a valley, it converges to the local optimum by the local search. As the number of starting points increases, the multiple starting point optimization has a higher probability to find the global optimum, but at the cost of the computational time. Besides, local optimization techniques, such as the conjugate gradient optimization method [21] and the Sequential Quadratic Programming (SQP) [17] need the gradients to drive the optimization. The computation of the gradient requires a large number of simulations. Also, for non-smooth objective functions, the traditional gradient based local search methods may stuck at non-smooth points.

Alternatively to deterministic optimization, researchers mainly used evolutionary computation algorithms (genetic algorithms, differential evolution, etc) for analog circuit optimization. Evolutionary Computation (EC) for global optimization mimic the biological mechanisms of evolution to approximate the global optimal solution of a problem [6]. In [22], a parallel genetic algorithm method is utilized for performance exploration. A global search explores a discretized version of the initial performance search space using a parallel genetic algorithm and generates a set of feasible performances. Each possible performance value represents a set of design variables. The proposed method suffers from a trade-off between the timing complexity and the accuracy of the search algorithm output. A non-uniform stochastic simulation using simulated annealing-based search is then employed to find the optimal sizing solution. In [1], the authors employ a genetic algorithm for simultaneous optimization of multiple performance parameters. The performances were evaluated using Support Vector Machine (SVM) [23] based models. However, SVM are black-box models. Thus, they are unable to reveal any qualitative aspects of the circuit behavior. In [2], the authors introduce the so-called Memetic Single-Objective Evolutionary Algorithm (MSOEA). The latter combines operators from the differential evolution and the genetic algorithm. It is specialized in handling large sizing problems with severe constraints. The sizing result of the mentioned works is very sensitive to various search parameters. It may often not meet the designer specifications. Also, the designer is frequently burdened with the task of tuning the optimizer parameters. Indeed, these techniques do not guarantee the non-existence of other possible good candidates of design parameters that satisfy the circuit specification.

An early attempt to use formal techniques in analog circuit sizing has been made in [24]. Using affine arithmetic, the authors calculate guaranteed bounds on the worst case behavior of the analog circuit and deterministically find the global optimum of the sizing problem by means of branch and bound optimization. Nevertheless, the feasibility of the method was demonstrated only on a small circuit.

Table 1.1 summarizes the above mentioned methods for analog circuits sizing. It describes the used evaluation techniques, the adopted optimization methods and the sized circuits.

Table 1.1: Summary of circuit sizing techniques

	Optimization Technique	**Performance Evaluation**	**Applications**
[25]	Parallel genetic algorithm and simulated annealing	Posynomial design equations and simulation	RF distributed amplifier Folded cascode amplifier
[26]	Convex optimization	Posynomial design equations and simulation	RF distributed amplifier
[17]	Sequential Quadratic Programming	Design equations and simulation	Ring operational amplifier Three-stage amplifier
[27] [19]	Convex optimization	Posynomial design equations	Two-stage amplifier
[20]	Polynomial optimization	Polynomial design equations	Two-stage amplifier Voltage controlled oscillator
[1]	Genetic algorithm	SVM-based models	Two-stage amplifier Cascode amplifier
[2] [28]	Genetic algorithm, differential evolution	Simulation	Folded cascode amplifier Telescopic cascode amplifier
[17]	Conjugate gradient optimization	Simulation	6T SRAM cell

Traditionally, analog circuit sizing methods use the width and length parameters of the transistors as design variables. In operating point driven (OPD) formulation [25], the circuit operating point is first selected then converted to transistor dimensions. In following, we briefly introduce the main techniques for OPD formulation and its advantages.

Operating Point Driven Circuit Sizing

The circuit operating point is a set of nodes voltages and the currents in the branches when the inputs to the circuit remain indefinitely at their quiescent values [29]. It is also known as bias point or quiescent point. Identifying the operating point is crucial because it directly affects the performance and yield of the circuit. In OPD circuit sizing, the circuit operating point is determined for a fixed transistor length value and the device sizes are computed out of it. When using this method, convergence problems often encountered in numerical simulation are avoided. Also, the design space is considerably reduced, as a proper choice of the device terminal voltages and current, ensures the correct operating region for the circuit. However, new analog sizing algorithms using the OPD technique have seldom been reported in recent years. One of the main reasons is that with the scaling down of the technologies, the transistor models are more complex. Consequently, available techniques for the conversion from currents and voltages to transistor sizes, such as DC root solving algorithms [30], local optimization [30], interpolation [31] and look-up table [32] based methods, face significant challenges on accuracy, efficiency and memory requirements. In this thesis, a novel approach for enabling the conversion from the bias to the size variables is proposed.

1.2.2 Yield Estimation

The *standard* approach to estimate the yield rate is the brute force Monte Carlo (MC) [33], which repeatedly draws samples from a predefined distribution of the process parameters and evaluates circuit performances with transistor-level SPICE simulation. MC has the advantages of simplicity and extremely general applicability. However, it can require very large numbers of expensive simulations for accurate yield estimation. MC is inefficient especially for circuits with rare failure events (e.g., static random access memories (SRAM)), because most of the samples fall into the feasible region, and only an extremely small fraction of samples are in the failure region [34].

Advanced *State of the art* MC for circuit yield analysis methods can be roughly divided into two categories: variance reduction techniques (e.g., Latin Hypercube Sampling (LHS) [35], Importance Sampling (IS) [36]) and low-discrepancy sequence-based methods (e.g., Quasi Monte Carlo (QMC) [37]). LHS partitions the range of each variable into non-overlapping intervals of equal probability and selects random values within each grid for every coordinate. By randomly combining the coordinate values, a set of latin hypercubes is constructed. Because of this stratification technique, the LHS method is capable of providing variance reduction of the yield estimation. However, it does not work much better than the conventional MC, especially for some problems that are difficult to be decomposed into a sum of univariate functions [37].

The key idea of IS based-methods is to shift the original probability density function (PDF) of the process parameters towards the most likely failure region. They have achieved remarkable speed-up when applied for the yield analysis of circuits characterized by rare failure event. However, IS lacks generality as it is designed for circuits with very high/low yield rate. Furthermore, generating the shifted/distorted PDF is often challenging and circuit specific, since this depends on the actual distribution of the circuit performance which is unknown beforehand.

Another critical issue of IS is that the proposed (i.e., shifted) sampling distribution may not cover effectively all failed samples when the circuit presents multiple disjoint failure regions induced by conflicting or multiple specification requirements [34]. Besides the multiple specication requirements, high-dimensional process variables also induce the multiple failure regions since the process parameters may have opposite influence on the performance metrics [36]. Only a few attempts have tackled the multiple failure regions case [38] [36]. In spite of that, while the method in [38] is applicable only to rare failure rate estimation in a very high-dimensional variation space (i.e., few hundreds), the authors in [36] reported that reduction techniques are required before applying their method for problems with more than 24 process parameters.

QMC is a popular approach that generates quasi-random numbers rather than purely-random samplings. It utilizes sample sets called Low Discrepancy Sequences (LDS), in which deterministically generated samples are uniformly distributed on the parameter space [37]. QMC methods are able to provide an improved integration error compared to LHS [37]. Yet, its convergence rate is found to be asymptotically superior to MC only for circuits with a moderate number of process parameters [35].

Other existing methods try to construct a surface boundary which separates the success and failure regions [39]. Once the boundary is constructed, the yield can be obtained by computing the volume of the failure region without circuit simulation. For low dimensional problems, this method can be efficient. However, such methods cannot handle high-dimensional problems with no more than three process variables. Even when considering only three process parameters, searching the whole failure boundaries in the parameters space is extremely complicated. The high-dimensional analysis (18∼24 process variables) is common and necessary in practical applications. Though, it makes the discrimination between failure and success regions by hypersurfaces very hard to achieve.

While above cited approaches present a variety of techniques to speed up and enhance the convergence of the traditional MC method, they fall short in addressing critical issues that can be summarized as follows:

- Optimally exploring the variational space that guarantees an acceptable accuracy and minimum computational time (i.e., a small number of transistor-level simulations).

- Scalability with respect to the process parameters size.
- Generality of application (i.e., handling different levels of yield rate, multiple performances metrics and multiple failure regions).

1.2.3 Yield Optimization

Yield optimization consists in finding the design point that has the largest margin from violating the specifications (i.e., maximum yield), when the circuit is subject to parameters variation. The search techniques reported in Section 1.2.1 (i.e., deterministic and stochastic search algorithms) can also be applied as optimization routines for yield optimization. In following, we discuss other available approaches that have been proposed particularly for yield optimization.

Most of the yield optimization methodologies are based on evolutionary computation algorithms. In [40] and [41], the authors employ Ordinal Optimization (OO) to allocate the simulation effort for each design point. At each optimization iteration, a sufficient computational budget is allocated for promising design points and a limited number of circuit simulation is employed to calculate the yield of non-critical solutions that have little effect on identifying the optimal design. In [40], OO is integrated with a two stage optimization strategy. The proposed algorithm uses differential evolution for global search and a random scale mutation operator for fine tunings to enhance the convergence speed of the yield optimization. In [41], OO is employed with multiple objective evolutionary algorithms. The optimization problem considers the yield, but it also ensures a trade-off between the yield and some other quality performance metrics. Both [40] and [41] present promising results in terms of computational cost and convergence speed. However, the accuracy of OO often cannot satisfy the requirement for objective optimization [6]. Furthermore, both cited methods require the fine tuning of numerous starting conditions. Also, they have to be run repeatedly due to their stochastic nature. In this thesis, we propose a deterministic optimization framework. The method does not require sophisticated knowledge for parameters fine-tuning. Also, a local optimization algorithm is incorporated to speedup the convergence.

Yield optimization methods include also device model corner-based methods and performance-specific worst-case design (PSWCD) methods. Device model corners-based methods check if the specifications are met at the extreme values of the process parameters. While computationally efficient due to the limited number of simulations required, different approaches have different choices to model corners which can be inaccurate or not realistic [6]. Also, the worst-case performance values are too pessimistic, as the corners correspond to the tails of the joint probability density function of the process parameters. Besides, corner-based methods account for global variation of the process parameters and do not include local variation effects which is critical in analog sizing. If the local variation is also considered, the number of simulations can be extremely large.

Worst-case optimization [42] denotes the task of finding the design point that minimizes the worst-case deviation of the performance from its nominal value. To do so, the lower and upper bounds of worst-case performances values as well as the corresponding design parameters are computed. This task is challenging and error prone as it is based on the linearization of the performances at the worst case design points, which is inaccurate especially in nanometer technologies [42].

Based on the above discussion and the stated limitations of the state-of-the-art, we propose in the next section a framework for yield-aware analog circuits sizing that tries to mitigate existing inefficiency issues.

1.3 Proposed Methodology

The main objective of this thesis is to develop a means to size robust analog circuits under process variation, for a given circuit topology. In particular, we target the sizing of analog cells (i.e., small to medium circuits having a basic operation). The first task towards our main goal is to determine a feasible subset of the design variables for which the circuit satisfies the specifications in nominal condition. Second, the best design solution in terms of robustness in light of parameters variation is selected. To do so, a yield estimation technique should efficiently evaluate the probability to satisfy the specification property, despite process variations. Moreover, an optimization engine selects the sizing solution with the highest yield rate. The framework for the proposed nominal circuit sizing and yield estimation and optimization is depicted in Figure 3.2. The proposed framework provides several novel techniques that address the limitations of existing yield-aware analog circuits sizing methods. It is composed of three complementary contributions: (1) a nominal circuit sizing approach based on SMT solving techniques; (2) an accelerated and reliable surrogate-based yield estimation; and (3) a yield optimization strategy. The three components of the methodology can be connected to produce the most robust sizing solution in light of process variation. However, each block can also be employed independently to perform its main functionality.

Given a set of circuit specification, a technology library and a circuit topology,

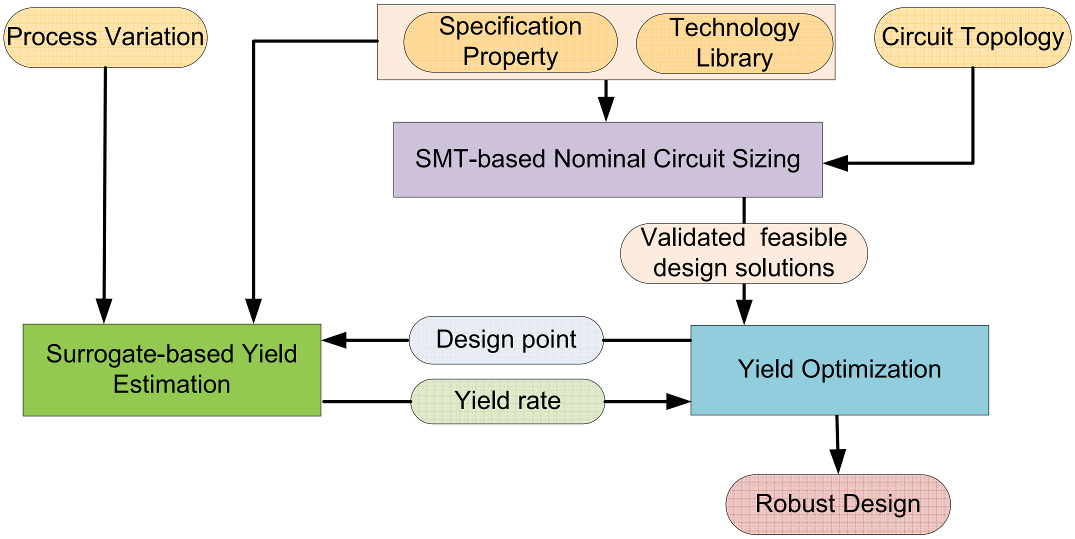

Figure 1.1: Overview of the Proposed Methodology

the nominal circuit sizing component computes a continuous set of validated feasible design solutions. The design solutions are guaranteed to satisfy the specification with high confidence in nominal condition. We use SMT solving techniques coupled with interval arithmetic to perform an exhaustive search of the design space. In order to efficiently use SMT technology, we employ a search space sampling approach and a parallel exploration to accelerate the sizing procedure.

Given the specification property and the technology library, the surrogate-based yield estimation block computes the yield rate of a design point under the effect of process variation. The design point can originate from the optimization process. However, the technique can also be applied independently to estimate the robustness of a given design point in the form of a SPICE netlist. The yield estimation technique combines the advantages of sparse regression and Satisfiability Modulo Theory (SMT) solving techniques. The method characterizes the failure regions as a collection of hyperrectangles in the parameters space. The yield computation is based on a geometric calculation of probabilistic volumes subtended by the located hyperrectangles.

The yield optimization stage takes as input the validated feasible design solutions and determines the most robust feasible design. The robust design maximizes the yield rate, despite process parameters variations. At each optimization iteration, a feasible design point is selected by the yield optimization engine and forwarded to the yield estimation block. The performances and yield rate are computed and fed back to the optimization engine. The optimization employs a two step exploration strategy. A global optimization phase locates a design point near the optimal solution that is used as a starting point by a local optimization phase. The local search constructs and optimizes local linear interpolating models of the yield to reach the global optimum with the highest yield rate.

We illustrate the application of each part of our methodology on various analog circuits to prove its effectiveness. We provide an in-depth analysis of our results and justify the use of various techniques proposed in this methodology. The nominal sizing stage and the surrogate-based yield analysis have been implemented via a link between MATLAB [43] and the SMT solver iSAT [14]. The optimization block is implemented in MATLAB. All simulations were performed using an 8-core Intel CPU i7- 860 processor running at 2.8 GHz with 32 GB memory and Linux operating system.

1.4 Thesis Contributions

The main objective of this thesis is the development of a methodology for enhancing analog circuit sizing in the presence of process variation. The pieces of this methodology can be integrated together in different ways to achieve the goal of sizing robust analog circuits. However, each of them has been used independently to perform its main functions and can also be adapted to any other related work. In the following, we list the main contributions of this work along with references to related publications provided in the Biography section that is given at the end of the thesis.

- Elaboration of a nominal circuit sizing methodology that computes a rough approximation of the design solution ranges as well as the space of feasible performances. The method is able to ensure an exhaustive coverage of the design search space and outputs guaranteed bounds on the feasible performance range [Bio-Cf2].

- Development of a novel method for fast and reliable computation of analog circuit yield that combines the advantages of sparse regression and Satisfiability Modulo Theory (SMT) solving techniques, and avoids issues in both. The yield estimation method is able to provide a guarantee on an exhaustive coverage of the circuit failure regions and hence tries to achieve reliable yield results [Bio-Jr1].

- Implementation of a novel method for analog yield optimization using a partition-based global search algorithm, which samples the most potential region of the feasible design space. A model-based local search is then integrated to highly speedup the convergence. Its efficiency is elevated by the reuse of existing simulation data of the global search phase [Bio-Cf1].

- The application of the proposed nominal circuit sizing and yield estimation and optimization techniques on various analog circuits including: a two-stage amplifier, an integrated ring oscillator, a 6T static RAM cell and a multi-stage fully-differential amplifier. These applications clearly demonstrate the feasibilities and the advantages of the diverse proposed methodologies.

1.5 Thesis Organization

The rest of the thesis is organized as follows: In Chapter 2, we detail our nominal circuit sizing methodology. We explain the formulation of the circuit sizing problem as a satisfiability problem and we describe the proposed SMT-based solving strategy that computes a continuous set of feasible design solutions. The usefulness of the proposed sizing technique is demonstrated with two analog circuits: a two-stage amplifier and a folded cascode amplifier for which we identify sets of continuous sizing solutions. After that, in Chapter 3, we describe the proposed surrogate-based yield estimation methodology in detail. We explain how we characterize the failure regions as a collection of hyperrectangles in the parameters space and how the yield estimation is based upon a geometric calculation of probabilistic volumes subtended by the located hyperrectangles. We also provide application results which prove that the proposed method is suitable for handling problems with tens of process parameters. Also, we demonstrate its effectiveness in handling circuits that usually require expensive run-time simulation during yield evaluation. In addition, in Chapter 4, we explain our new optimization technique applied for yield optimization. and show its effectiveness through the analysis of two CMOS amplifiers under the effect of process variations. Finally, Chapter 5 provides concluding remarks and several directions for future research.

Chapter 2

SMT-based Nominal Circuit Sizing

In this chapter, we focus on analog circuit sizing in nominal condition. We present an approach for enhancing the sizing procedure using Satisfiability Modulo Theory (SMT). The circuit sizing problem is encoded using nonlinear constraints. An SMT-based algorithm exhaustively explores the design space, where the biasing-level design variables are conservatively tracked using a collection of hyperrectangles. The device dimensions are then determined by accurately modeling the geometry-level design parameters as a function of the biasing-level design parameters. We demonstrate the feasibility and efficiency of the proposed methodology on a two-stage amplifier and a folded cascode amplifier.

2.1 Circuit Sizing Methodology

Given a set of specifications, a circuit topology and a technology library, design constraints are derived and input to an SMT-based circuit sizing step. During this stage, technology information are first collected in order to characterize transistor parameters. For this purpose, we use extensive circuit simulations to infer polynomials that approximate the small signal parameters of n-MOS and p-MOS transistors as a function of biasing voltages and currents. The analytical-based performance expressions are formulated using the constructed models. Given a well-defined set of specifications and the circuit topology, the design constraints are derived and input to an SMT-based design space exploration algorithm. This step uses interval arithmetics with SMT solving techniques to ensure a complete coverage of the design space. The output of this block is an over-approximation of each device operating points as well as the feasible performance space.

The next step consists in converting the continuous ranges of operating point of each device into interval-valued transistor dimensions. For that, we use simulation and clustering to fit a piecewise polynomial approximation that relates the transistor width parameter to the biasing voltages and currents. The model efficiently captures the nonlinear function that relates multidimensional scattered data generated using analog simulation.

The goal of the last step is to verify if the circuit satisfies the feasible performance space given the generated ranges of devices sizes. For that, Monte Carlo simulation is performed at the circuit level. If the requirement in terms of accuracy is met, the method outputs continuous ranges of validated feasible sizing solutions. Otherwise, the design constraints can be further investigated and the SMT solver parameters adjusted. Figure 2.1 summarizes our analog circuit sizing method using satisfiability modulo theory (SMT) techniques.

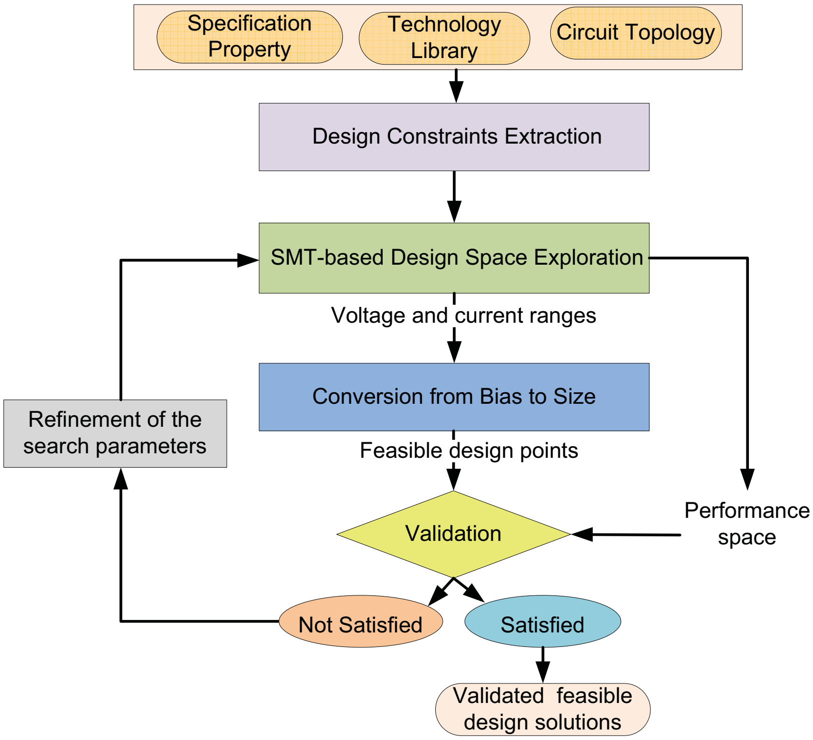
Specification Property
Technology Library
Circuit Topology
Design Constraints Extraction
SMT-based Design Space Exploration
Voltage and current ranges
Conversion from Bias to Size
Feasible design points
Refinement of the search parameters
Validation
Performance space
Not Satisfied
Satisfied
Validated feasible design solutions

Figure 2.1: Circuit sizing methodology

2.1.1 Design Constraints Extraction

At the beginning of our circuit sizing methodology, a characterization of the transistor small signal parameters is performed. For this purpose, a relational model that relates each small signal parameter (g_m, g_{ds}) to the biasing-level design variables of the transistor (I_{ds}, V_g, V_d, V_s) is constructed (Figure 2.2). First, small signal parameters and biasing-level design variables of n-MOS and p-MOS transistors are swept during Monte Carlo simulation in SPICE [7] using the Latin Hypercube Sampling (LHS) [44]. Then, only feasible variables ensuring that the transistors are biased in saturation or in triode are retained. All training pairs of transistor operating points and small signal parameters are then formulated as a least square error problem and fed into a third order polynomial regression step to determine the fitting parameters. High degree polynomials are avoided to prevent prohibited complex equations and ill-conditioning. The extracted models can be reused multiple times for a given technology which ensures the generality of our approach. The problem formulation can be written as follows:

$$\min_{\alpha} \sum_{\mathbf{n=1}}^{\mathbf{N}} (\mathbf{y_n} - \mathbf{f}(\mathbf{x_n}, \alpha))^2$$

where $\mathbf{y}$ is the transistor small signal parameter and $\mathbf{x}$ is the set of biasing variables values obtained from circuit level simulation, $f(\mathbf{x}, \alpha)$ represents the regression model, α the fitting parameters and N the number of data samples.

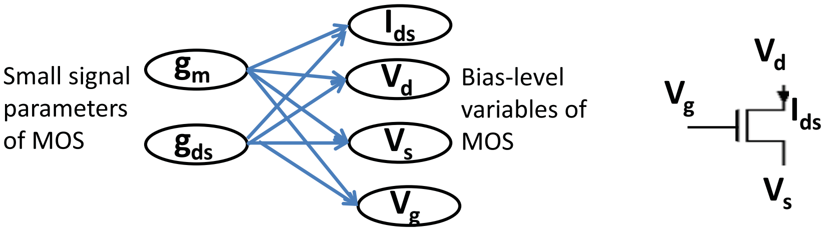

Figure 2.2: Mapping small signal parameters to biasing-level variables of MOS

The performance equations are expressed as a function of biasing-level transistor variables and/or small signal parameters. The SMT problem is a conjunction of the initial space of each design variable, the performance equations, the specifications and other design constraints, such as restricting the transistors to operate in the saturation/triode region, symmetry constraints and Kirchhoff's Current Law (KCL).

In general, the problem formulation can be written as follows:

$$
\begin{aligned}
\mathbf{X_{min}} \leq \quad & \mathbf{X} \quad \leq \mathbf{X_{max}} \\
\mathbf{Y_{min}} \leq \quad & \mathbf{Y} \quad \leq \mathbf{Y_{max}} \\
\mathbf{Z_{min}} \leq \quad & \mathbf{Z} \quad \leq \mathbf{Z_{max}} \\
\mathbf{y_j} \quad = \quad & \mathbf{f_j}(\mathbf{X}, \alpha_\mathbf{j}) \\
\mathbf{z_p} \quad = \quad & \mathbf{g_p^{(k)}}(\mathbf{X}, \mathbf{Y}) \\
\mathbf{k}(\mathbf{X}) \quad \oplus \quad & \mathbf{0}
\end{aligned}
\tag{2.1}
$$

- $\mathbf{X} = \{\mathbf{x_i}, i = 1 \ldots l\}$ are the biasing-level design variables.
- $\mathbf{Y} = \{\mathbf{y_j}, j = 1 \ldots m\}$ are the transistors small signal parameters.
- $[\mathbf{X_{min}}, \mathbf{X_{max}}]$ are the ranges of the biasing-level design variables.
- $[\mathbf{Y_{min}}, \mathbf{Y_{max}}]$ are the ranges of the small signal parameters.
- $\mathbf{y_j} = \mathbf{f_j}(\mathbf{X}, \alpha_\mathbf{j}), j = 1 \ldots m$, are the mapping equations from $\mathbf{X}$ to $\mathbf{Y}$ and $\alpha_\mathbf{j}$ the fitting parameters.
- $\mathbf{Z} = \{\mathbf{z_p}, p = 1 \ldots P\}$ are the performance metrics, (e.g., gain, bandwidth, etc.) and P is the number of performance metrics.
- $[\mathbf{Z_{min}}, \mathbf{Z_{max}}]_\mathbf{p}$, $p = 1 \ldots P$, are the boundary values specifications of the performance metrics z_p.
- $\mathbf{z_p} = \mathbf{g_p^{(}}\mathbf{X}, \mathbf{Y})$ are the performance equations of the p^{th} performance metric.
- $\mathbf{k}(\mathbf{X}) \oplus \mathbf{0}$ are the set of device matching constraints and transistor operation conditions, KCL, where $\oplus$ stands for $=, \leq, \geq, <,$ or $>$.

2.1.2 Design Space Exploration

The aim of the design space exploration is to determine the feasible performance as well as the transistors operating points ranges given the sizing constraints *constr* and a set of specifications $[Z_{min}, Z_{max}]_p$. Our approach, which we explain in the sequel, is summarized in Algorithm 2.1.

Alg. 2.1. SMT-based design space exploration

```
Require: $S, P, constr, [Z_{min}, Z_{max}]_p$
 1: $X_f = \emptyset$, $Z_f = \emptyset$, $N_S = S^P$
 2: for all $ind = 1 \rightarrow N_S$ do in parallel
 3:    $z_p \subseteq [z_{pmin}, z_{pmax}]_{ind}$
 4:    repeat
 5:       Invoke iSAT($constr$)
 6:       if a candidate is found then
 7:          Invoke INTLAB($constr, candidate$)
 8:          if Locate box then
 9:             $X_f \leftarrow X_f \cup Xbox$
10:             $Z_f \leftarrow Z_f \cup Zbox$
11:             Update($z_p, Zbox$)
12:          end if
13:       end if
14:    until Unsatisfiable
15: end for
16: return  $Z_f$: Feasible performance space
            $X_f$: Biasing-level design variable space
```

The cost of solving nonlinear SMT problems increases exponentially with the problem dimension. It would be then infeasible to run the search over a large initial space of performance. For these reasons, we propose first to split the problem into $N_S = S^P$ subproblems that we solve simultaneously (Line 2). Each subproblem is limited to a possible combination of performance boundaries. That is, for each subproblem, a possible combination of the performance metrics is traversed $z_p \subseteq [z_{pmin}, z_{pmax}]_{ind}, p = 1 \ldots P$ (Line 3). For example, if the circuit requires two performance metrics ($P = 2$) with $S = 5$ descritazation steps (i.e., sampling density), then the overall combinations of performance space to be explored is $N_S = S^P = 5^2$. Obviously, we can observe that the complexity increases with more specs and greater precision in sampling. However, a parallel enhancement is adopted to reduce the timing complexity. Figure 2.3 summarizes our proposed design space exploration scheme.

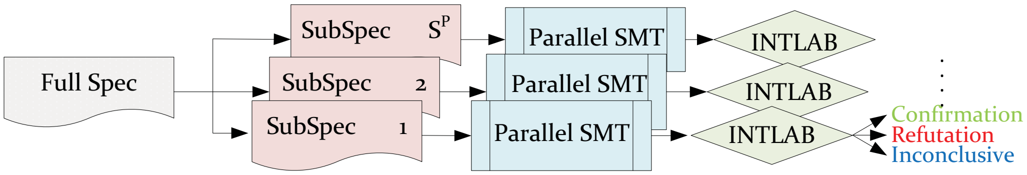

Figure 2.3: Parallel design space exploration

The SMT solver [14] returns a set of continuous ranges (*candidate*) of each variable (Line 6). However, the set of interval solutions is only an over-approximation that can be devoid of any real solution to the constraints. The uncertainty can be alleviated by setting a high resolution of the returned candidate. Still, this will dramatically increase the computation time. Owing to this, the size of the interval solution (resolution) is adjusted on the fly for a trade-off between computational cost and over-approximation. Also, for each set of intervals proposed by the SMT solver, we use the MATLAB toolbox for interval arithmetic INTLAB [45] to further refine the solution (Line 7). Given the candidate solution as interval initial condition and the sizing equations, INTLAB either refutes the existence of any solution or produces a hyperbox that is contained in the candidate region and guaranteed to contain the solution (Line 8). Though, INTLAB may also fail to either confirm or refute the existence of a solution. One possible reason of this non-determinism case is that the *candidate* returned by the SMT solver may contain multiple roots. In this case, the hyperrectangle can be returned to the solver to be further analyzed.

The feasible performance space $Zbox$ and the devices operating points $Xbox$ are then merged into Z_f and X_f, respectively (Lines 9 and 10). The function *Update* removes *Zbox* from the search space by adding the constraint $Zbox \not\subseteq z_p$. This will force the solver to search for new solutions [46]. Finally, when all reachable hypercubes are found, the solver will return *Unsatisfiable*, providing a guarantee on complete coverage of the search space. In fact, the SMT solver provides a guarantee on unsatisfiability. An unsatisfiable result means that there is no additional *candidate* solution to the sizing problem constraints.

2.1.3 Conversion from Bias to Size

The aim of this step is to allow the conversion from device operating point to device size. For this purpose, we first propose to construct a model $\hat{f}_w$ of the transistor width parameter as a function of the branch current and the node voltage for n-MOS and p-MOS transistors. We next present an approach that approximates the width parameter as piecewise polynomial model over a number of regions, which is illustrated in Figure 2.4.

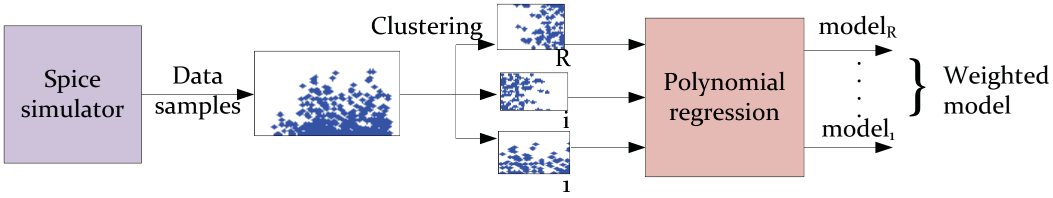

Figure 2.4: Clustered polynomial regression

We use k-means clustering to subdivide the multidimensional scattered data of branch current, node voltage and width data samples, generated using Monte Carlo simulation in SPICE, into R regions. The number of clusters is set to an initial guess as the first and is updated after that to guarantee the accuracy of the model and a practical time required to generate it. The result of the clustering procedure is a discrete version of the data. Each region is represented by its centroid x_r. A polynomial model of third order is then generated at each region using multivariate nonlinear regression. The regression problem for each region $r = 1 \ldots R$, can be written as a least square minimization problem as shown in Equation 2.2.

$$\min_{\beta_\mathbf{r}} \sum_{\mathbf{n=1}}^{\mathbf{N}} (\mathbf{w_r^{(n)}} - \hat{\mathbf{f}}_\mathbf{r}(\mathbf{x^{(n)}}, \beta_\mathbf{r}))^\mathbf{2} \tag{2.2}$$

where w_r and x are, respectively, the set of width and bias voltage and current values obtained from circuit level simulation, N the number of samples, $\hat{f}_r(x, \beta_r)$ represents the regression model that approximates w_r and β_r is a set of fitting parameters. To avoid overfitting even more, a weighted model evaluation is proposed. The value of the weight function $weight_r$ should be close to one when the vector of bias values x approaches the centroid x_r, and should attenuate to zero when x leaves x_r. We propose to choose a Gaussian function [47] where $\sigma = 0.01$ is a predefined constant as given in Equation 2.3.

$$\begin{gathered} \hat{\mathbf{f}}_\mathbf{w} = \sum_{\mathbf{r=1}}^{\mathbf{R}} \mathbf{weight_r} * \hat{\mathbf{f}}_\mathbf{r} \\ \mathbf{weight_r} = \mathbf{e}^{\frac{-(\mathbf{x_r}-\mathbf{x})}{\sigma}} \end{gathered} \tag{2.3}$$

Once the macromodel $\hat{f}_w$ is generated, the next step consists in determining the transistor size ranges $[w_{min}, w_{max}]$, given the set of transistor operating points X_f. Algorithm 2.2 provides a description of the global optimization (GO) based approach. For each transistor (i from 1 to n), the algorithm calculates, using $\hat{f}_w$, the minimum and the maximum of the transistor width $[w_{imin}, w_{imax}]$, when its bias voltages and current x_i are constrained to $[X^i_{f(min)}, X^i_{f(max)}]$, as well as, the transistor operating in the appropriate operation region. The search algorithm alg is the interior-point method [48] and x_0 is a well-defined starting point.

Alg. 2.2. Transistor width range computation

```
Require: $\hat{f}_w, w_0, X_f, n, alg$
for $i \in 1\ to\ n$ do
    $x_0 = (X^i_f(max) - X^i_f(min))/2$
    $[w_{imin}, w_{imax}] = GO(\hat{f}_w(x), x_0, alg) subject\ to\ x_i \in [X^i_{f(max)}, X^i_{f(min)}]$
end for
return $[w_{min}, w_{max}]$
```

2.1.4 Validation

The goal of this step is to verify wether the circuit performances, when fed to the circuit simulator, are within the performance space Z_f or close with an acceptable level of error. Then, Monte Carlo simulation is performed over the ranges of sizes $[w_{imin}, w_{imax}]$ to compute the reachable performances. In case the level of accuracy is not acceptable, we refine the discretization resolution (i.e., solution size). However, the timing complexity is sacrificed as trade-off. The inaccuracy can also raise from the fitting error. This can be targeted by increasing the data samples or the order of the polynomials models.

2.2 Applications

In this section, we present the application of our circuit sizing technique on the example of a two-stage amplifier [49] and a cascode amplifier [6]. In what follows, all the used circuits are in $0.18\mu m$ technology based on BSIM3 models [50]. The lengths of all transistors are kept constant and set to $0.36\mu m$. However, the small signal parameters models and the width parameter model can be constructed for different constant values of transistor length. Therefore, it is possible to set different values of the transistors lengths. The transistors widths are allowed to vary from $1\mu m$ to $40\mu m$ and the current from $1\mu A$ to $8.4mA$. The approach has been implemented via a link between MATLAB and the SMT solver iSAT.

2.2.1 Modeling Evaluation

In this part, we evaluate our modeling technique explained in Section 2.1.3. The performance of the proposed clustered polynomial regression is compared to four multidimensional regression approaches [51] available in MATLAB, for the same n-dimensional test case as shown in Table 2.1. In this comparison, we consider 10000 Monte Carlo data samples of transistor channel width, currents and voltages, 75% of them for training and 25% for testing. We model the channel width as a function of the transistor operating points for n-MOS and p-MOS. The prediction ability of each regressor is tested by calculating the normalized mean square error (NMSE=$\frac{\|\hat{f}-w\|_2^2}{\|w\|_2^2}$).

Table 2.1: Test Error (NMSE) Comparison

MOS	CPR	PR	NN	SVM	MARS
n-MOS	$0.48\,10^{-2}$	$9.3\,10^{-2}$	2.2	-	$7.1\,10^{-2}$
p-MOS	$0.37\,10^{-2}$	$9.6\,10^{-2}$	2.5	-	$6.2\,10^{-2}$

While the least-squares Support Vector Machines (SVM) was not able to converge to an exact solution and were running indefinitely, the Neural Networks (NN) are too inaccurate (test error > 200%) and hence are not a suitable regressor for high-dimensional test case. With the Multivariate Adaptive Regression Splines (MARS) using piecewise cubic sampling, the error is less than 10%, while for our clustered polynomial regression (CPR) approach, it is below 1% and 10 times less than using polynomial regression (PR) without clustering. Combining data clustering with local multivariate polynomial approximation is a robust means of approximating the non-linear function that relates multidimensional scattered data. The model takes a few seconds to be constructed and can be reused multiple times for the same technology and MOS model. In general, the modeling error should be less than 2×10^{-2} to ensure the accuracy of the sizing result.

2.2.2 Two-stage Operational Amplifier

We consider a two-stage amplifier [49] as shown in Figure 2.5. The load capacitance is set to $10pF$. In order to guarantee the stability of the amplifier, we set the constraint $C_c > \frac{g_{m1}}{g_{m6}} 1.7 C_L$ [49]. Appropriate operating regions are ensured by imposing saturation constraints on each transistor. Matching relations were also imposed: $V_{d2} = V_{d1}$, $I_1 = I_2 = 0.5 I_5$, $I_5 = I_8$. The analytical expression of the performance metrics can be approximately expressed as given in Equation 2.4.

$$\begin{aligned} Av &= \frac{g_{m1} g_{m6}}{(g_{ds2} + g_{ds4})(g_{ds6} + g_{ds7})} \\ P_{DC} &= v_{dd}(I_8 + I_5 + I_6) \\ GBW &= \frac{g_{m1}}{2\pi C_c} \\ g_{mi} &= \hat{g}_{mi}(V_{di}, V_{gi}, V_{si}, I_i, \alpha_i), i = 1, 6 \\ g_{dsi} &= \hat{g}_{dsi}(V_{di}, V_{gi}, V_{si}, I_i, \beta_i), i = 2, 4, 6, 7 \end{aligned} \tag{2.4}$$

where $V_{di}, V_{gi}, V_{di}, I_i$ refer to the operating point of of the transistor M_i, $\hat{g}_{mi}$ and $\hat{g}_{dsi}$ its small signal parameters and α_i, β_i the fitting parameters. Column 2 of Table 2.3

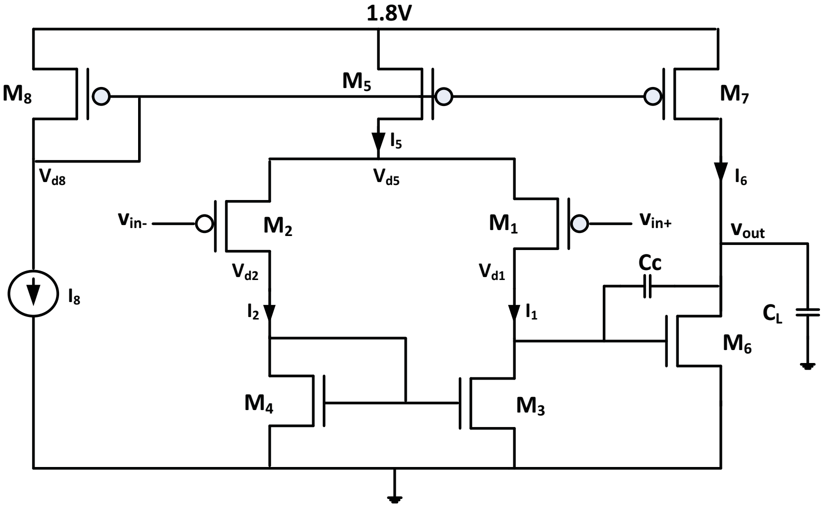

Figure 2.5: Two-stage amplifier

reports the performance specifications on the gain Av, gainbandwidth GBW, power P_{DC}, phase margin PM and the input common mode range $ICMR$. Our aim is to study the feasibility of the specification and determine the reachable performance space and devices sizes.

In order to show the effectiveness of our approach in speeding up the search process, we divide the SMT search problem into different numbers of subproblems and compare their run-times as shown in Table 2.2. In fact, the run-time tends to decrease linearly when the number of subproblems increases. A minimum run-time can be reached with a sampling density S equal to 4. In this case, the three performance boundaries ($P = 3$) related to the gain, gainbandwidth and power constraints are subdivided into $S^P = 4^3 = 64$ possible combinations of performance boundaries. The total combinations are explored and a speedup of ×10 is achieved. The speedup comes from: (1) the reduction of the search space allowed by the problem subdivision; and (2) the capability of producing multiple satisfiable solutions simultaneously thanks to the parallel implementation.

Table 2.2: Two-stage amplifier experimental results

Samples	Run-time [s]	# Candidate regions	# Solutions	# Spurious regions
S=1	660	226	205	21
S=2	340	220	206	14
S=3	290	216	201	15
S=4	61	221	205	16

We report the number of candidate regions computed by iSAT in Column 3 of Table 2.2. The number of solutions confirmed by INTLAB is shown in Column 4. The spurious regions are those reported by iSAT but not confirmed by INTLAB. That is, the solver cannot derive any contradictions within the reported candidate regions, with respect to the constraints and the adopted resolution (i.e., solution size). It does not mean, however, that the box actually contains a solution. Thus, the presence of spurious regions is explained. The number of these regions is simply the total number of reported candidate regions minus the number of solutions confirmed by INTLAB.

Table 2.3: Specification and results of our method

Perf metrics	Specifications	Our method	MC
$Av(dB)$	[60, 70]	[60, 66.5]	[59.7, 66]
$GBW(MHz)$	$[2, 6]$	[2.05, 3.62]	[2.5, 3.6]
$P_{DC}(mW)$	$[0.09, 0.17]$	[0.12, 0.17]	[0.12, 0.18]
$ICMR(V)$	[0.8, 1.6]	[0.8, 1.3]	-
$PM(°)$	≥ 60	-	[128, 135]

The performance space computed by our SMT design space exploration method is reported in Column 3 of Table 2.3. The proposed methodology outputs continuous ranges of design variables, as shown in Table 2.4. In order to evaluate the accuracy of our results, Monte Carlo (MC) simulation has been run with 1000 trials where the design variables are uniformly distributed over the computed sizing ranges shown in Table 2.4. Not surprisingly, our sizing results are guaranteed to fulfill the generated feasible performance with a small violation as the device models used in the formulation of the constraints do not totally match the sophisticated models utilized in the validation step. Still, the violation is very small owing to the accuracy of the extracted models.

Table 2.4: Design variables ranges of the two-stage amplifier

Design variables	Ranges
$w_1 = w_2(\mu m)$	[7.1, 8.95]
$w_3 = w_4(\mu m)$	[2.9, 3.15]
$w_5 = w_8(\mu m)$	[9.4, 9.6]
$w_6(\mu m)$	[5.4, 5.6]
$w_7(\mu m)$	[7.12, 8.1]
$I_8(\mu A)$	[25.3, 35.2]
$C_c(pF)$	[7.5, 8.1]

We compare our results with an optimization-based method using Genetic Algorithm (GA) [1] applied for the sizing of the two-stage amplifier circuit in $0.18\mu m$ technology. The goal of GA is to simultaneously optimize Av, GBW and the circuit phase margin (PM) and to search for the candidate solution that achieves the best trade-off. The achieved performances are reported in Table 2.5 and the total computation time is 437.63 *sec.*

Table 2.5: Specification and results of GA [1]

Perf metrics	Specification	GA	SPICE
$Av(dB)$	maximize	61.8	61.7
$GBW(MHz)$	maximize	3.21	2.75
$PM(^\circ)$	maximize	145	122

The performances of the optimized circuit are verified using SPICE simulation. Our method is able to locate higher performances when compared to the optimal design solution computed by GA. The search ability of our SMT-based approach obviously outperforms the GA-based method thanks to an exhaustive and complete coverage of the large design space, as well as, an accurate modeling of the circuit characteristics. Another strength of our method is the computation of a continuous safe subset of design variables for which the circuit satisfies the specifications, while the optimization based-method GA does not have this ability, as it only targets a nominal design point. We offer valuable information about the performances bounds when the design variables are subject to variation.

2.2.3 Folded Cascode Amplifier

We consider a folded cascode amplifier circuit [49] as shown in Figure 2.6. The inputs voltages (V_{in-}, V_{in+}) are set to $0.9V$ and the load capacitance is fixed to $5pF$. Appropriate saturation constraints are imposed on each transistor. The symmetry constraints are applied as follows: $V_{d2} = V_{d1}$, $V_{d7} = V_{out}$, $V_{d6} = V_{d13}$, $I_1 = I_2 = 0.5I_3$, $I_{10} = I_9$, $I_{13} = I_6$ and $I_3 = I_4$. The expressions of the performance metrics are given in Equation 2.5. The design specifications are shown in Column 2 of Table 2.7.

$$\begin{aligned}
P_{DC} &= v_{dd}(I_6 + I_{13} + I_3 + I_4) \\
GBW &= \frac{g_{m2}}{2\pi C_L} \\
SR &= \frac{I_3}{C_L} \\
Av &= g_{m2} R_{out} \\
R_{out} &= \frac{g_{m11}}{g_{ds11}}\left(\frac{1}{g_{ds2}+g_{ds10}}\right) \parallel \frac{1}{g_{ds13}} + \frac{1}{g_{ds12}}\left(1+\frac{g_{m12}}{g_{ds13}}\right)) \\
g_{mi} &= \hat{g}_{mi}(V_{di}, V_{gi}, V_{si}, I_i, \alpha_i), i = 2, 11, 12 \\
g_{dsi} &= \hat{g}_{dsi}(V_{di}, V_{gi}, V_{si}, I_i, \beta_i), i = 10, 11, 12, 13
\end{aligned} \tag{2.5}$$

The run-time for determining the performance space and the continuous range

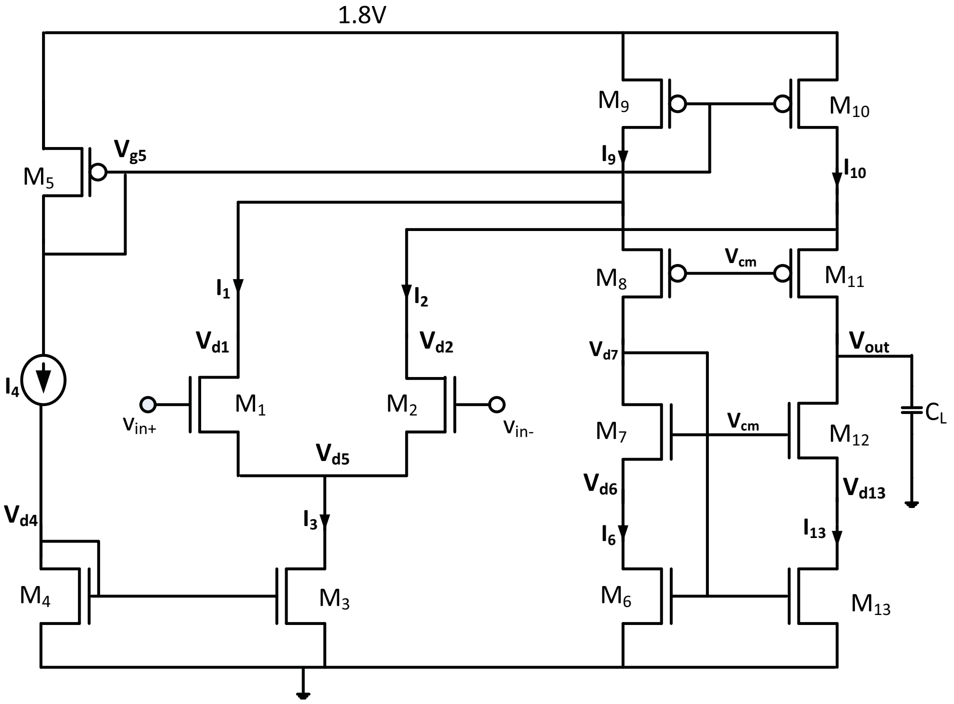

Figure 2.6: Folded cascode amplifier

of operating points with different sampling density is reported in Table 2.6. In fact, a minimum run-time of $90s$ is reached with $S = 4$ showing significant speedup of $\times 10$ when compared to the naive approach ($S = 1$). This result indicates the capability of our approach in reducing the design space exploration computational time and improving the efficiency in solving the SMT problem.

Table 2.6: Cascode amplifier experimental results

Samples	Run-time [s]	#Candidate regions	#Solutions	#Spurious regions
S=1	960	186	176	10
S=2	440	181	175	6
S=3	290	188	176	12
S=4	90	190	174	16

The circuit sizing methodology computes continuous ranges of design variables, as shown in Table 2.8. The reported ranges are the validated feasible design solutions.

The cascode amplifier satisfies its specifications, with high guarantee of accuracy, for any sizing solution included in these ranges. The reachable performance space is reported in Column 3 of Table 2.7. These performance boundaries are computed by the SMT-based design exploration stage. They represent an approximation of all possible performance values that can be reached when the design variables are constrained to the validated feasible design solutions. We also include the results of 1000 Monte Carlo (MC) simulations (Column 4) where the design variables are uniformly distributed over the computed sizing ranges shown in Table 2.8. Our method successfully identifies the true feasible design solutions with high confidence.

Table 2.7: Specification and results of our method

Perf metrics	Specification	Our method	MC
$Av(dB)$	[60, 70]	[60, 65]	[61.3, 67.5]
$GBW(MHz)$	[80, 90]	[80, 83]	[79, 84]
$P_{DC}(mW)$	$[1, 1.29]$	[1.25, 1.28]	[1.24, 1.27]
$SR(V/\mu s)$	$[60, 75]$	[64, 65.6]	[61.2, 63]

Table 2.8: Design variables ranges of the folded cascode amplifier

Design variables	Ranges
$w_1 = w_2(\mu m)$	[30.1, 39.9]
$w_9 = w_{10}(\mu m)$	[11.15, 11.21]
$w_8 = w_{11}(\mu m)$	[16.41, 16.53]
$w_7 = w_{12}(\mu m)$	[2.9, 5.8]
$w_6 = w_{13}(\mu m)$	[3.1, 5.75]
$w_3 = w_4(\mu m)$	[13.4, 13.51]
$w_5(\mu m)$	[18.2, 18.31]
$I_4(\mu A)$	[320, 328]
$V_{cm}(V)$	[0.750, 0.751]

We compare our experimental results with high-ability optimization algorithms including Genetic Algorithm with penalty function (GA), Differential Evolution (DE) algorithm and Memetic Single Objective Evolutionary Algorithm (MSOEA), which were employed to size the cascode amplifier circuit in $0.18\mu m$ technology. The results are summarized in Table 2.9. For the above three methods, the objective is to minimize the power while satisfying the constraints in Column 2 of Table 2.9. The evaluation of the performances is accomplished in the circuit simulator HSPICE [2]. While GA and DE fail to find feasible solutions even with multiple different sets of search parameters and initial conditions [2], our method is guaranteed to determine a range of continuous design parameters when they exist. Indeed, less power consumption P_{DC} is achieved while better quality of gain Av, slew rate SR and GBW are successfully located when compared to MSEOA. This result is accomplished thanks to an exhaustive parallel design space exploration and a good coverage of the design space. We also ensure minimal area occupation which may vary from $61.2\mu m^2$ to $72\mu m^2$ while it is $426.34\mu m^2$ for MSEOA.

Table 2.9: Specification and results of [2]

Perf metrics	Specification	GA	DE	MSOEA
$Av(dB)$	≥ 60	61.89	60	60.12
$GBW(MHz)$	≥ 80	3.13	51.13	80
$P_{DC}(mW)$	minimize	0.03	0.74	1.29
$SR(V/\mu s)$	≥ 60	1.56	33.97	60.03
Run-time (s)	-	173	161	185

Unlike these search algorithms that return one local solution to the sizing problem, our approach determines a continuous safe subset of the design parameters that is guaranteed to comply with the specifications while it is often computationally expensive and time consuming to size a circuit such that it obeys properties over a range of design parameters. Moreover, our SMT-based search technique highly relieves the sizing solution from the uncertainty inherited from optimization-based method. Besides, it can be applied to any circuit and does not require special problem formulation.

2.3 Summary

In this chapter, we presented a methodology for characterizing a feasible region of the sizing solution space. Given the circuit topology and the specification properties, we compute a rough approximation of the design solutions in nominal condition. The proposed scheme is an alternative approach to the existing analog sizing techniques (optimization-based) with additional trust and better coverage of the search space. However, in real designs, we are facing inevitable variations in the parameters of the manufacturing process such as thickness of the oxide layer and threshold voltage. The next chapter presents a method that evaluates the robustness of a design solution in the presence of parameters variation.

Chapter 3

Surrogate-based Yield Estimation

In this chapter, we propose a new method for accelerated and reliable computation of parametric yield that combines the advantages of sparse regression and Satisfiability Modulo Theory (SMT) solving techniques, and avoids issues in both. The key idea is to characterize the failure regions as a collection of hyperrectangles in the parameters space. Towards this goal, the method constructs a sparse polynomial models based on adaptive LASSO (Least Absolute Shrinkage and Selection Operator) [52] to find low degree approximations of the circuit performances. A procedure inspired by statistical model checking is then introduced to assess the model accuracy. Given the constructed models, an SMT-based solving algorithm is employed to locate the failure hyperrectangles in the parameters space. The yield estimation is based on a geometric calculation of probabilistic volumes subtended by the located hyperrectangles. We demonstrate the effectiveness of our method using circuits that require expensive runtime simulation during yield evaluation. They include: an integrated ring oscillator, a 6T static RAM cell and a multi-stage fully-differential amplifier.

3.1 Yield Rate Estimation Methodology

Before presenting the proposed methodology for surrogate-based yield estimation, we briefly explain our main objective and define terms that will be used in the rest of the chapter. Suppose that $p = [p_1, p_2, \ldots, p_l]$ is a l-dimensional continuous random variable modeling process variations. Such random variables include the variations of gate length ΔL, oxide thickness Δt_{ox} and threshold voltage ΔV_{th}, etc., associated with each circuit device. Without loss of generality, we further assume that the random variables in the vector p are mutually independent and follow a truncated normal distribution with $\pm 3\sigma$ and zero mean. We define the parameters (i.e., variation) space P as the set of all possible combinations of the random variables. In general, the yield rate can be mathematically represented as:

$$\mathbf{Y}^* \;=\; 1 - P_f = 1 - \int_{\Omega} pdf(p)dp \tag{3.1}$$

where $pdf(p)$ is the joint probability density function of p and Ω denotes the failure region, i.e., the region of the parameters space where the performances are not satisfied (can be a single region or multiple disjoint regions). We denote the integral in Equation 3.1 to be the probabilistic hypervolume of Ω [53]. Figure 3.1 is a geometrical illustration in two dimensions.

In general, the multidimensional integral in Equation 3.1 cannot be directly computed since the failure region Ω usually establishes a complex nonlinear integration

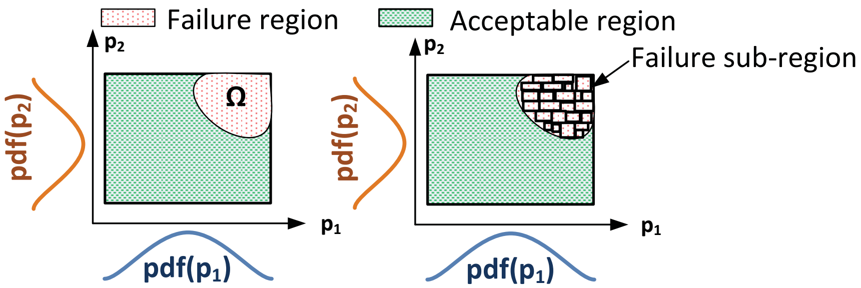

Figure 3.1: 2-D parameters space

boundary. In our method, we propose to characterize Ω as a collection of high dimensional sub-regions (i.e., hyperrectangles). The probabilistic hypervolume of each sub-region is then evaluated and employed to estimate the total yield. Obviously, the accuracy of the yield estimation depends strongly on how well the sub-regions are approximated. In this chapter, we will mainly focus on this characterization problem and develop novel algorithms to make it tractable and computationally efficient.

The methodology in Figure 3.2 details the proposed approach for yield estimation. The technique takes as input a design point derived from a yield optimization block, as described in the proposed framework in Chapter 1. It can also be applied independently to estimate yield rate of a design point described as a SPICE netlist.

First, an adaptive sparse regression technique is applied to extract surrogate models of the circuit performances. In order to optimize the modeling step, a dimension reduction technique keeps the most significant process parameters. The proposed algorithm sorts the process parameters by weight assignment and prunes the unimportant parameters. Then, a low-degree and sparse polynomial model of each circuit performance is constructed based on adaptive LASSO. The LASSO method assigns adaptive weights for penalizing the coefficients of the polynomial terms and yields a consistent estimate of the model coefficients. The model is iteratively built until the

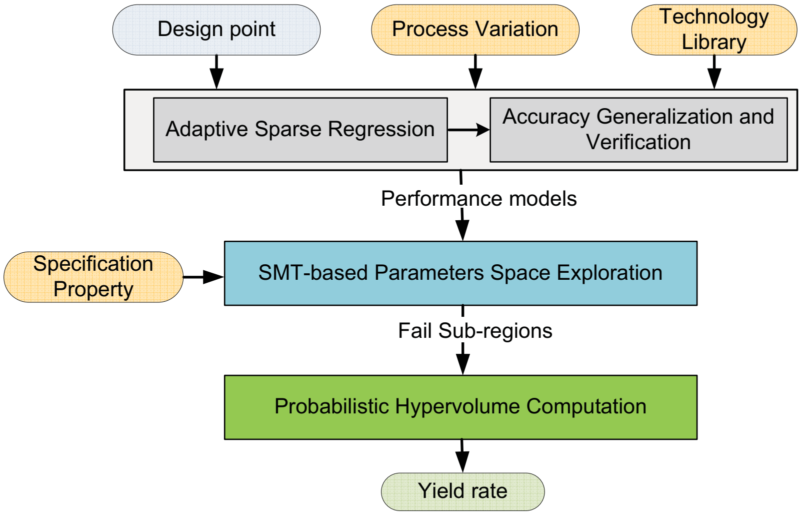
Design point
Process Variation
Technology Library
Adaptive Sparse Regression
Accuracy Generalization and Verification
Performance models
Specification Property
SMT-based Parameters Space Exploration
Fail Sub-regions
Probabilistic Hypervolume Computation
Yield rate

Figure 3.2: Yield estimation methodology

requirement in terms of accuracy is met. A procedure inspired by statistical model checking is then introduced to verify the model accuracy for a chosen confidence level. The resulting model can be viewed as a statistically guaranteed approximation of the circuit behavior. The subset of the circuit response space where each performance of interest does not meet the specification is conservatively characterized as a set of intervals. Based on the extracted models, SMT solving is not employed to compute the exact failure sub-regions in the parameters space. Instead, it is used to find only an over-approximation of them. The integration of interval arithmetics to remove the undesirable over-approximation, trades off between the computational cost and the conservativeness of SMT. A parallel exploration of the failure performance space allows the simultaneous finding of multiple satisfiable solutions and significantly speeds up the search process. Finally, the yield is estimated based on the probabilistic hypervolumes of the failure sub-regions. The methodology outputs an estimation of the circuit yield rate (or its equivalent percentage) that is the probability that it satisfies its specification under the effect of process variation.

3.1.1 Adaptive Sparse Regression

In this section, we seek to produce an accurate surrogate model using polynomials with structured sparsity. The modeling technique should be performed with a minimum number of circuit simulations. Besides, the model must be computationally efficient (i.e., not complex) and hence tractable for the subsequent SMT solving stage.

Pre-sampling and dimension reduction

The goal of pre-sampling is to approximately sketch the circuit behavior. We use the LHS method in the parameters space to generate a set of training samples. Given n training samples, we denote $X = [x_1, x_2, \ldots, x_n]$ an $l \times n$ matrix, where each sample $x_i = [p_{i1}, p_{i2}, \ldots, p_{il}]$ is an l-dimensional vector. Next, transistor level SPICE simulation is performed to evaluate the performance metric using these samples. We denote $Y = [y_1, y_2, \ldots, y_n]$ the n observations of the property, i.e., the value of the circuit response we seek to fit.

The parameters reduction maps the high dimensional process parameters space to a lower-dimensional space. We leverage the Regressional ReliefF (RReliefF) [54] algorithm to prune the process parameters and to select a smaller number of features. The algorithm uses samples based learning to assign a relevance weight to each parameter. Each feature weight reflects its ability to perturb the circuit response. The quality estimate ranges in $[-1, 1]$. Equation 3.2 [54] shows the weight updating formula for each feature of the process parameter vector p.

$$\begin{aligned}
\mathbf{V(p)} &= \mathbf{W(p)} + \frac{\mathbf{N_{dCdp}}}{\mathbf{N_{dC}}} - \frac{(\mathbf{N_{dp}} - \mathbf{N_{dCdp}})}{\mathbf{n} - \mathbf{N_{dC}}} \\
\mathbf{N_{dp}} &= \frac{|\mathbf{value(p, x_i)} - \mathbf{value(p, x_j)}|}{\mathbf{max(p)} - \mathbf{min(p)}}\mathbf{d(i, j)} \\
\mathbf{N_{dC}} &= |\mathbf{y_i} - \mathbf{y_j}|\mathbf{d(i, j)} \\
\mathbf{N_{dCdp}} &= |\mathbf{y_i} - \mathbf{y_j}||\mathbf{value(p, x_i)} - \mathbf{value(p, x_j)}|\mathbf{d(i, j)}
\end{aligned} \tag{3.2}$$

RReliefF starts with a l-long weight vector, V, of zeros, and iteratively updates V for all features in p. This process is repeated for the total number of instances n. In each iteration, the algorithm randomly selects a sample x_i and finds all k nearest samples x_j around x_i, in terms of Euclidean distance. The relevance level of each feature is then assigned by approximating the terms in Equation 3.2, where N_{dp} is a normalized difference between the values of parameters in the vector p for the two instances x_i and x_j. The quantity $d(i, j)$ [54] takes into account the distance between samples by assigning greater weight to closer samples, and N_{dC} corresponds to the difference between the performances of the two samples. The term N_{dCdp} quantifies the probability that two nearest samples have different performances and different values of parameter. The weight increases if the circuit responses of nearest samples differ and decrease in the reverse case. In practice, a feature is relevant when the weight is strictly positive and irrelevant in the opposite case [55]. The algorithm only requires $O(lnlog(n))$ time, and is noise-tolerant and robust to feature interactions. Besides, in difference to the partial-derivative based sensitivity analysis, RReliefF is more robust as it avoids the instability of numerical methods.

Adaptive least-squares regression using LASSO

Once the most relevant process parameters are captured, we seek to construct a surrogate model of each performance metric involved in the circuit specification. The performance function is a local perturbation around its nominal value. We use polynomial basis which are very often used to approximate such a local variation [56]:

$$\mathbf{f}(\mathbf{p}) \simeq \sum_{\mathbf{m}=\mathbf{1}}^{\mathbf{M}} \mathbf{c_m g_m}(\mathbf{p}) \tag{3.3}$$

where f is a smooth circuit performance approximated as a linear combination of M basis functions, c_m are the model coefficients and $g_m(p)$ is a basis functions (linear, quadratic or cubic polynomials). The unknown model coefficients c_m are determined by solving a set of linear equations at a number of sampling points (training data), which is usually solved as a least squares problem:

$$\min_{\mathbf{c_{m,m\in[1,M]}}} \|\mathbf{f}(\mathbf{p}) - \mathbf{q}(\mathbf{p})\|_{\mathbf{2}}^{\mathbf{2}}, \quad \mathbf{q}(\mathbf{p}) = \sum_{\mathbf{m}=\mathbf{1}}^{\mathbf{M}} \mathbf{c_m g_m}(\mathbf{p}) \tag{3.4}$$

In fact, the number of process parameters is often large, while the number of training samples is greatly limited by the computational cost. Given the limited computational budget, the underlying system is rank deficient. Therefore, the solution c_m (i.e., the vector containing unknown model coefficients) is not unique and impossible to identify without additional constraints. To solve this problem, we propose to employ adaptive LASSO as a weighted regularization technique for simultaneous consistent estimation and variable selection [52]:

$$\min_{\mathbf{c_{m,m\in[1,M]}}} \|\mathbf{f}(\mathbf{p}) - \mathbf{q}(\mathbf{p})\|_{\mathbf{2}}^{\mathbf{2}} + \alpha \sum_{\mathbf{m}=\mathbf{1}}^{\mathbf{M}} \|\frac{\mathbf{c_m}}{\mathbf{w_m}}\|_{\mathbf{1}} \tag{3.5}$$

where α is a nonnegative regularization parameter. $\|\|_1$ stands for the l_1-norm of a vector which denotes the sum of the absolute values of all elements in the vector. The weighted penalty function $\alpha \sum_{m=1}^{M} \|\frac{c_m}{w_m}\|_1$ is an additional constraint that forces the coefficients c_m to behave regularly by shrinking the coefficients towards 0 as α increases. Data-dependent weights w are employed for penalizing different coefficients in the l_1 penalty. By allowing relatively higher penalty function (higher weight) for small coefficients and lower penalty function (lower weight) for larger coefficients, the adaptive LASSO neutralizes the influence of the coefficient magnitude on the l_1 penalty function. Thus, it reduces the coefficient estimation bias compared with the standard LASSO. Furthermore, the adaptive LASSO shrinkage retains the attractive convexity property of the standard LASSO [52]. Most importantly, it is proved to be near-minimax optimal [57]. The weight w can be any consistent estimate of c_m. Here, we select $w = (X^T X)^{-1} X^T Y$ to be the ordinary least square estimate of c_m [57], where X^T denotes the vector transpose of X.

An overview of our proposed surrogate modeling scheme is shown in Figure 3.3. In particular, our implementation starts by selecting the most important predictors and applies adaptive sparse regression in a stepwise fashion. The idea is that higher degree terms are included only when necessary to avoid high order model. As long as the model accuracy satisfies the convergence condition, the training process stops so that the model is easier to interpret and more efficient to evaluate.

Algorithm 3.1 provides a simplified description of the adaptive sparse regression algorithm. This algorithm is applied to construct a surrogate model $q(\tilde{p})$ of each performance metric intervening in the circuit specification. It requires as inputs a set of training X and test samples X^t and their corresponding circuit responses Y and Y^t, respectively. Typically, the number of training samples can be selected from

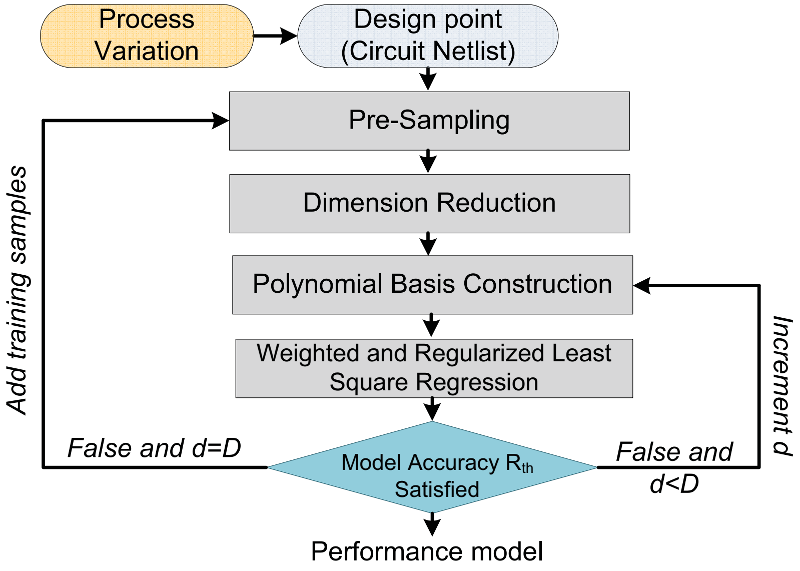
Process Variation
Design point (Circuit Netlist)
Pre-Sampling
Dimension Reduction
Polynomial Basis Construction
Weighted and Regularized Least Square Regression
Model Accuracy R_{th} Satisfied
Performance model
Add training samples
False and d=D
False and d<D
Increment d

Figure 3.3: Surrogate model training

200 to 500 while the test samples from 100 to 300. In Line 1, we use the RReliefF algorithm to select a smaller number of features $\tilde{p}$ and filter out features that hardly have contributions to the circuit response.

Alg. 3.1. Response surface-based surrogate model training

```
Require: $X$, $X^t$: Data samples, $Y$, $Y^t$ : Circuit response,
    $D = 3$: Maximum degree, $d = 0$, $k = 15$, $R_{th}$: Accuracy threshold
 1: $\tilde{p} \leftarrow$RReliefF$(X, Y, k)$,
 2: $X_{\tilde{p}} \leftarrow$ select$(X, \tilde{p})$, $X^t_{\tilde{p}} \leftarrow$ select$(X^t, \tilde{p})$
 3: while $d < D$ and $\varepsilon > R_{th}$ do
 4:     $d \leftarrow d + 1$
 5:     $\tilde{X}_f \leftarrow$ expand_polynomial_basis $(X_{\tilde{p}}, \tilde{p}, d)$
 6:     $w \leftarrow$ compute_weight$(\tilde{X}_f, Y)$
 7:     $q(\tilde{p}) \leftarrow$ adaptive_lasso$(w, \tilde{X}_f, Y)$
 8:     $\varepsilon \leftarrow$ verify$(q, X^t_{\tilde{p}}, Y^t)$
 9: end while
10: if $\varepsilon \leq R_{th}$ then
11:     Return (Accuracy model met!)
12: else
13:     Generate fresh samples and go to 5
14: end if
```

The parameter k is the number of nearest instance considered by RReliefF [43]. In all experiments conducted in this chapter (cf. Section 3.2), we find that $k = 15$

provides stable and reliable reduction results. In Line 2, the function *select* extracts the observation $X_{\tilde{p}}$ and $X^t_{\tilde{p}}$ corresponding to the reduced process parameters space $\tilde{p}$ from the original set X and X^t, respectively. Then, the algorithm operates in an iterative fashion. At each iteration, the polynomial degree is incremented (Line 4). The idea is that higher degree terms are included only when necessary to avoid high order models. In Line 5, we construct a set of polynomial basis $g_m(\tilde{p})$ of degree d. The polynomial terms of $g_m(\tilde{p})$ are obtained by expanding all the terms in the d-degree polynomial $(1 + p_1 + \cdots)^d$. Then, $\tilde{X}_f$ maps the reduced data matrix $X_{\tilde{p}}$ to each expansion terms of $g_m(\tilde{p})$. In Lines 6 and 7, the weights w are computed and the adaptive LASSO problem in Equation 3.6 is solved using the coordinate descent algorithm [43].

$$\min_{\mathbf{c_{m,m\in[1,M]}}} \|\mathbf{Y} - \tilde{\mathbf{X}}_{\mathbf{f}}\mathbf{c_m}\|_{\mathbf{2}}^{\mathbf{2}} + \alpha\|\frac{\mathbf{c_m}}{\mathbf{w_m}}\|_{\mathbf{1}} \tag{3.6}$$

The coordinate descent iterations terminate when the relative change in the size of the estimated coefficients drops below $1e^{-9}$. It is important to note that c_m are computed each time the degree d is incremented. This re-calculation is required because the new basis function constructed at the current iteration step may change the model coefficient values calculated at previous iteration steps. The regularization parameter α is chosen during the training process. It is selected such that it minimizes an estimate of expected prediction error based on 10 fold cross-validation applied to the training samples. In Line 8, the test samples $X^t_{\tilde{p}}$ are used to verify the accuracy of the current trained model. The prediction ability of the model is tested by calculating the normalized mean square error (NMSE=$\frac{\|q(X^t_{\tilde{p}})-Y^t\|_2^2}{\|Y^t\|_2^2}$). When the error of the performance model ε is less than a given threshold, named R_{th}, or the degree d reaches the limit D, the iteration stops.

If the desired accuracy is not met and d reached the maximum degree D, then fresh samples are generated and added incrementally to the training sample set as long as the model accuracy does not satisfy the convergence condition (Line 13). The generation of the fresh samples uses a triangulation approach as explained in [36]. How to select the parameter R_{th} will be discussed in Section 3.2.4.

Compared with previous techniques for modeling analog performances using LASSO [56], our proposed method has two main advantages: (1) It has two levels of reduction that makes the modeling problem tractable. First, it identifies significant parameters. Then, selects the appropriate basis functions from a large pool of possible polynomial candidates; (2) it tackled the issue of dependence on magnitude of LASSO by penalizing more heavily larger coefficients in the l_1 norm; and (3) the training scheme is designed to extract a low degree polynomial that can be efficiently handled in the SMT solving step.

In practice, the number of samples required to compute ε cannot be fixed in advance. If a very large evaluation set is employed to evaluate the error ε, then the resulting model accuracy can be trusted. However, this would prohibitively increase the computational cost. Next, we propose to employ statistics to provide a certain confidence level on the model accuracy with a probability of error which can be prespecified.

3.1.2 Accuracy Generalization and Verification

While the surrogate model error ε can never be totally eliminated, its accuracy verification is primordial to prove the reliability of the yield estimation methodology. The surrogate model accuracy $(1-\varepsilon)$ can be considered φ-guaranteed if :

$$\forall \mathbf{p}, \tilde{\mathbf{p}} \in \mathbf{P}, \quad \mathbf{Pr}((\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) \quad \geq \varphi \tag{3.7}$$

where Pr and err stand for probability and model error, respectively. In other words, the model error is at most ε for at least φ portion of the parameter space. Clearly, at this stage there is no guarantee on the model accuracy $(1-\varepsilon)$. The purpose of this step is to determine a generalized accuracy under the process parameter space, given a probability/level of confidence φ.

To do so, we employ and extend the statistical procedure proposed by Younes [58] that regards the model checking of a system as a hypothesis testing problem and solves it using Walds sequential probability ratio test (SPRT) [59]. The idea is to check the accuracy property in Equation 3.7 on a samples set of simulations and to decide whether the model $q(\tilde{p})$ satisfies the property based on the number of executions for which the property holds compared to the total number of executions. With such an approach, we do not need to explore and test all possible values of process parameters. We rather answer the question of whether the model satisfies the property with a probability greater than or equal to a value $\varphi \in [0,1]$. Furthermore, we propose a simple, yet elegant modification to the SPRT test which allows the computation of a generalized model accuracy ε. The problem is treated based on two exclusive hypothesis testing given as follows:

$$\begin{aligned} \mathbf{H_0} &= \mathbf{Pr}(\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) \geq \varphi + \delta = \varphi_{\mathbf{2}} \\ \mathbf{H_1} &= \mathbf{Pr}(\mathbf{err}(\mathbf{f}(\mathbf{p}), \mathbf{q}(\tilde{\mathbf{p}})) \leq \varepsilon) < \varphi - \delta = \varphi_{\mathbf{1}} \end{aligned} \tag{3.8}$$

where H_0 and H_1 are known as the null and the alternative hypothesis and 2δ forms a small region called the indifference region [58], on both sides of the cutting point φ. If the probability is between φ_1 and φ_1 (the indifference region), then we say that the probability is sufficiently close to φ so that we are indifferent with respect to which of the two hypotheses is accepted. The method determines on the fly the number of simulations needed to achieve a desired accuracy and provides a convenient way to control the trade-off between precision and computational cost. To decide between the two hypothesis, the test proceeds by computing at the n^{th} stage of the test, i.e., after making n observations, a log likelihood ratio given as [59]:

$$\mathbf{\Lambda_n} = log\frac{\prod_{i=1}^{n} z_{\varphi_1}(b_i)}{\prod_{i=1}^{n} z_{\varphi_2}(b_i)} = log\frac{\int_0^{\varphi_1}\prod_{i=1}^{n} z^{b_i}(1-z)^{1-b_i}dz}{\int_{\varphi_2}^{1}\prod_{i=1}^{n} z^{b_i}(1-z)^{1-b_i}dz} \tag{3.9}$$

where n represents the total number of samples or the test length, $b_1, b_2, \cdots, b_n$ is a collection of Bernouilli random variables denoting the outcome of the accuracy property (Equation 3.7) with random samples $x_1, x_2, \cdots, x_n$ drawn from the parameters space. $z_{\varphi_1}(b_i)$ and $z_{\varphi_2}(b_i)$ are the probability mass function of the Bernouilli distribution parameterized by φ_1 and φ_2, respectively. The quantity $\mathbf{\Lambda_n}$ is finally given as:

$$\mathbf{\Lambda_n} = log\frac{B_{\varphi_1}(k+1, n-k+1)}{A - B_{\varphi_2}(k+1, n-k+1)} \tag{3.10}$$

where $0 \leq k \leq n$ is the number of successful inequality test, $A = \frac{1}{(n+1)C_k^n}$ and $B_{\varphi 1}$ and $B_{\varphi 2}$ are the incomplete Beta functions. H_0 is accepted if $\Lambda_n \leq a$ and H_1 is accepted if $\Lambda_n \geq b$, where $a = \log(\frac{\alpha}{1-\beta})$ [59] and $b = \log(\frac{1-\alpha}{\beta})$ [59]. α and β are two decision error rates that determine the strength of the test, where α is the type I error rate or false positive and β is the type II error rate or false negative.

The procedure is summarized in Algorithm 3.2. It repeatedly checks the accuracy

Alg. 3.2. Verification and generalization of the model accuracy

```
Require: $q$: Surrogate model, $\varepsilon$: model error, $\tilde{p}$,$p$: Process parameters, $\varphi_1, \varphi_2$: Probabilities, $\alpha, \beta$:
    Error rates.
 1: $a = \log(\frac{\alpha}{1-\beta}); b = \log(\frac{1-\alpha}{\beta})$, $X^t_{\tilde{p}}, Y^t$
 2: $n = 0$; $k = 0$;
 3: while $a < \Lambda_n < b$ do
 4:     $n \leftarrow n + 1$
 5:     $x_n \leftarrow$ Sample the parameters space $P$
 6:     $f \leftarrow$ Simulate the circuit at the parameters $x_n$ and measure $f$
 7:     $X^t_{\tilde{p}}, Y^t \leftarrow$Update($X^t_{\tilde{p}}, Y^t, x_n, f$)
 8:     if err($q(X^t_{\tilde{p}}), Y^t$)$> \varepsilon$ then
 9:         $\varepsilon \leftarrow$ err($q(X^t_{\tilde{p}}), Y^t$)
10:     else
11:         $k \leftarrow k + 1$
12:     end if
13:     Evaluate $\Lambda_n(n, k, \varphi_1, \varphi_2)$
14: end while
15: if $\Lambda_n \leq a$ then
16:     Accept $H_0$
17: else
18:     Accept $H_1$
19: end if
```

property with fresh samples x_n drawn from the parameters space p (Line 5). After measuring the sample response f (Line 6), we add the fresh observation (x_n, f) to the testing samples $(X^t_{\tilde{p}}, Y^t)$ (Line 7) and we compute the normalized mean square error (Line 8). We say that the inequality test is a success if the property holds, and a failure otherwise. Upon each success, we increment the counter k (Line 11) and continue with fresh samples until a failure occurs. In this case, we update and generalize the error ε (Line 9). We can therefore characterize the required number of observations as $inf\{n, \Lambda_n \notin]a, b[\}$. Clearly, this number increases if α and β are smaller but also if φ is very close to one. We provide in Section 3.2.4 a discussion concerning these parameters.

3.1.3 SMT-based Parameters Space Exploration

The objective of this stage of the methodology is to exhaustively probe the parameters space and to determine failure hyperrectangles, i.e., regions where the circuit fails to satisfy the design specification. Our approach is summarized in Algorithm 3.3. In order to conservatively find the reachable parameters values, we formulate the SMT problem *constr* as a conjunction of the space of the process parameters, the constructed surrogate models and the specification violation constraints. In general, the problem can be formulated as:

$$\begin{aligned}
\mathbf{p^{min}} &\leq \mathbf{p} \leq & \mathbf{p^{max}} \\
\mathbf{f_k(\tilde{p}_k)} &= & \mathbf{q_k(\tilde{p}_k)} \\
\mathbf{f_K^{min}} &\leq \mathbf{f_K} \leq & \mathbf{f_K^{max}, K = 1} \\
\bigvee_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{f_k^{min}} &\leq \mathbf{f_k} \leq & \mathbf{f_k^{max}, K > 1}
\end{aligned} \tag{3.11}$$

where $f_k(\tilde{p}_k), k = 1 \ldots K$, are the performance equations, K is the total number of performance metrics involved in the design specification, $\tilde{p}_k$ is the reduced process parameters set associated to the k^{th} performance metric. $[p^{min}, p^{max}]$ are the ranges of the process parameters determined from their probabilities distributions, where $p = [p_1, p_2, \ldots, p_r]$ and $r = dim(\cup_1^k \tilde{p}_k)$ is the dimension of the reduced parameters space. As mentioned before, we use a truncated normal shape to model the process parameters. If $\pm 3\sigma$ variation is considered then, the upper and lower bounds of the process parameters p^{min} and p^{max}, respectively, are defined as:

$$\mathbf{p^{min} = p^{nom} - 3\sigma; p^{max} = p^{nom} + 3\sigma} \tag{3.12}$$

where p^{nom} is a vector of nominal values. $[f_k^{min}, f_k^{max}]$ are the bounds that approximate the failure region of the circuit operation in the performance space. For example, if we are given an oscillator circuit designed at a nominal frequency f_{nom} and the maximum allowed frequency deviation is $\bigtriangleup f$, then the $failure$ frequency region is defined as: $[f^l, f_{nom} - \bigtriangleup f[\cup]f_{nom} + \bigtriangleup f, f^u]$, where f^l and f^u are the minimum and maximum performances values reached by the circuit. It is important to set a conservative approximation of f^l and f^u in order to let the solver discover any possible failure of the circuit response under the defined parameters variation. The over-conservativeness is especially necessary for circuits with rare failure event where the circuit simulation in the initial pre-sampling cannot be sufficient to sketch the performance bound. We provide in Section 3.2.4 a discussion concerning the setting of the failure bound.

In case of multiple performance metrics, the specification violation is mathematically formulated as a disjunction of failure performance bounds, as given in Line 4 of Equation 3.11, where $\bigvee$ denotes the logical OR operator. In fact, a high dimensional region in the parameters space is considered as a failure region if any performance metric involved in the specification is not satisfied.

Alg. 3.3. SMT-based parameters space exploration

```
Require: $S, K, constr, N_S = S^K$
for all $ind = 1 \rightarrow N_S$ do in parallel
    $f_k \subseteq [f_k^{min}, f_k^{max}]_{ind}$
    repeat
        Invoke iSAT3($constr$)
        if a $candidate$ is found then
            Invoke INTLAB($constr, candidate$)
            if Locate $p^{box}$ then
                Return($Perf^{box}, p^{box}$)
                Update($Perf^{box}, f_k$)
            end if
        end if
    until $Unsatisfiable$
end for
```

The SMT solver iSAT3 is known to attempt to solve NP-complete problems. Solving these problems, in their worst case, would take time which is exponential in the number of variables to solve. It would be then infeasible to run the search over a large initial space of $failure$ performance bounds $[f_k^{min}, f_k^{max}]$. For these reasons, we propose first to split the SMT problem *constr* into $N_S = S^K$ subproblems that we solve simultaneously (Line 1 of Algorithm 3.3). For example, if the circuit requires two performance metrics ($K = 2$) with $S = 5$ uniform descretazation steps, then the overall combinations of the performance space to be explored is $N_S = S^K = 5^2$. Each subproblem is limited to a possible combination of performance boundaries. More precisely, for each subproblem, a possible combination of the failure regions in the performance space is traversed and the specification violation constraint is formulated as: $\bigvee_{k=1}^{K} f_k \subseteq [f_k^{min}, f_k^{max}]_{ind}, k = 1 \ldots K$. Also, it is important to note that all subproblems have the same SMT constraints and the same process parameters variables. Based on this, solving all subproblems is completely equivalent to solving the original SMT problem.

Obviously, we can observe that the complexity increases with more performance metrics and greater precision in sampling. For this reason, the SMT subproblems are solved in parallel to reduce the timing complexity. The solver returns a set of continuous ranges of each variable (i.e., a hyperectangle) in the SMT constraints (Line 5). However, the set of interval solutions is only an over-approximation (*candidate*) that can be devoid of any real solution to the constraints. The uncertainty can be alleviated by setting a high resolution of the returned *candidate*. Still, this will dramatically increase the computation time. Owing to this, the size of the interval solution (resolution) is adjusted for a trade-off between computational cost and over-approximation.

We only use the SMT solver to refine the initial search space towards a *candidate* solution and to discard the infeasible solution. Afterwards, for each set of intervals proposed by iSAT3, we employ INTLAB to further refine the *candidate* solution (Line 6). Given the *candidate* solution as interval initial condition and the performance equations, INTLAB either refutes the existence of any solution in the candidate solution returned by the SMT solver or produces a hyperrectangle p^{box} that is contained in the *candidate* region and guaranteed to contain the solution (Line 7). The widths of the interval solution p^{box} returned by INTLAB are smaller than the *candidate* region proposed by the SMT solver.

The result of the refinement process is a set of interval process parameters p^{box} and its corresponding reachable performances $Perf^{box}$ (Line 8). The function $Update$ in Line 9 removes $Perf^{box}$ from the search space by adding the constraint $Perf^{box} \not\subseteq f_k$. This will force the solver to search for new solutions. Finally, when all reachable hyperrectangles are found, the solver will return *Unsatisfiable*, providing a guarantee on a complete coverage of the search space (i.e., the failure region). In fact, Algorithm 3.3 exploits the strength of the SMT solver (i.e., its search space coverage capabilities) while avoiding its disadvantages.

3.1.4 Yield Estimation

In the previous stage of the methodology, we have characterized Ω as a set of high dimensional sub-regions in the parameters space: $\Omega \simeq \{p^{box}\}_{1 \longrightarrow n_f}$, where n_f is the total number of located sub-regions. A failure sub-region is a hyperrectangle that is modeled as a cartesian product of orthogonal intervals $p^{box} = ([p_1^l, p_1^u] \times \ldots \times [p_r^l, p_r^u]])$. We recall that the parameters p are assumed independent and continuous random variables. The probability that the process parameters fall into a single sub-region

p^{box} is estimated in two dimensions (for illustrative purposes) as:

$$\begin{aligned} P(p_1, p_2 \in p^{box}) &= \int_{p^{box}} pdf(p)dp = \prod_{i=1}^{2} P(p_i^l \leq p_i \leq p_i^u) \\ &= \prod_{i=1}^{2} CDF(p_i^u) - CDF(p_i^l) \end{aligned} \tag{3.13}$$

where P stands for probability, $p_1^u, p_1^l, p_2^u, p_2^l$ are the coordinates of the sub-region in two dimension (as shown in Figure 3.4), and $CDF(p_i)$ [43] represents the cumulative distribution function of p_i.

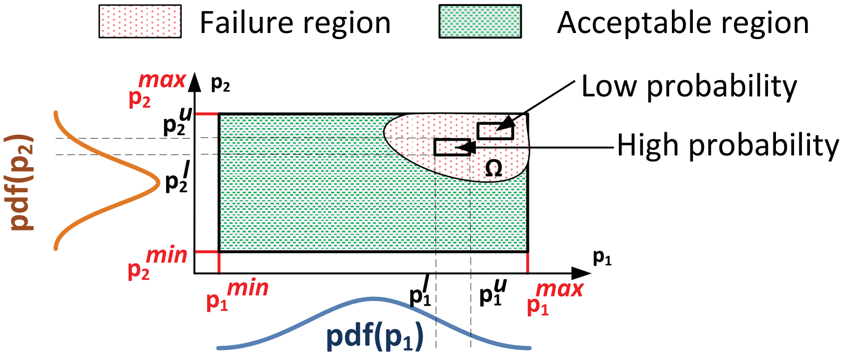

Figure 3.4: Illustration of the coordinates of a failure sub-region in 2-D parameters space

For the total n_f failure sub-regions in r-dimensional parameters space, the probability that the design constraints are satisfied in the presence of parameters variation is generalized as:

$$\begin{aligned} \mathbf{Y}^* &= 1 - P_f = 1 - \sum_{j=1}^{n_f} \int_{\{p^{box}\}_j} pdf(p)dp \\ &= 1 - \sum_{j=1}^{n_f} [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_j \end{aligned} \tag{3.14}$$

The multidimensional integral in Equation 3.14 is the probabilistic hypervolume of a single sub-region. Obviously, the contribution of a located sub-region to the failure probability P_f is higher when the coordinates of the hyperectangles are closer to the center of the process parameters space. The circuit yield is computed according to Algorithm 3.4.

Alg. 3.4. Yield rate computation

```
Require: $\{p_{box}\}_{1 \longrightarrow n_f}$
    $P_f = [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_1$
1: for all $j = 2 \rightarrow n_f$ do
2:     $p_j^{box} \leftarrow p_j^{box} - \bigcap(p_j^{box}, p_{1 \rightarrow j-1}^{box})$
3:     $P_f \leftarrow P_f + [\prod_{i=1}^{r} CDF(p_i^u) - CDF(p_i^l)]_j$
4: end for
5: $\mathbf{Y}^* \leftarrow 1 - P_f$
```

In Line 2, the hyperrectangle is refined for more precision and accuracy. The term $\bigcap(p_j^{box}, p_{1 \rightarrow j-1}^{box})$ is the region resulting from the overlapping between the located boxes. The overlay may occur if some hyperrectangles share the same values of process parameters or due to the conservativeness of interval arithmetic computation.

3.2 Applications

In this section, we present the application of the yield rate estimation methodology described in the previous section on the examples of a three-stage ring oscillator, a six transistor SRAM cell and a three-stage operational amplifier (op-amp). In the experiments, the circuits are designed in a commercial TSMC (Taiwan Semiconductor Manufacturing Company) 65 nm process [60] and simulated in HSPICE with BSIM4 transistor models. The local mismatch variables are considered as the process parameters including the oxide thickness $\triangle t_{ox}$, threshold voltage under zero bias $\triangle V_{th}$, channel width $\triangle w$ and channel length $\triangle L$. We assume that each process parameter follows a truncated normal distribution. We use the statistical device models offered by the IC foundry TSMC. We use the transistor mismatch model [60] with $Vdd = 1V$ and standard threshold voltage. The mismatch model uses principal component analysis (PCA) [52] to model the process parameters as a set of independent random variables. Given a set of correlated random variables $p^{'}$, PCA is applied to find a set of independent random variables p that represent the original correlated random variables $p^{'}$: $p = Tp^{'}$. The linear transformation matrix T is determined such that p is modeled as a function of mutually independent and standard Normal (i.e., zero mean and unit variance) random variables. The random variables in p are called principal components. The essence of PCA can be interpreted as a coordinate rotation of the space defined by the correlated random variables in $p^{'}$.

The algorithms parameters are selected as follows. The value of the convergence condition R_{th} in Algorithm 3.1 is selected as $2.\,10^{-2}$. We also choose a degree limit D of 3 for all performances models. For the model verification step, we use $\varphi = 0.95$, a symmetric test strength $\alpha = \beta = 0.01$ and an indifference region of size 10^{-3}, indicating that the statistical test covers at least 95% of the parameter space with a high statistical condence. The choice of these parameters values is discussed in Subsection 3.2.4.

3.2.1 Three-stage Ring Oscillator

We consider a three-stage ring oscillator [49] as shown in Figure 3.5. The lengths of all transistors are fixed to $65nm$. The width of all p-MOS transistors is $3\mu m$. The width of all n-MOS transistors is $2.5\mu m$. The oscillation frequency is chosen to be the performance metric of interest. The nominal frequency f_{nom} is 3.207 GHz calculated via periodical steady state (PSS) simulation. The design specification requires that the variation of the frequency should be within 2.5% of f_{nom}. The oscillation frequency is affected by various process parameters in the transistors. The local mismatch variables of each transistor are considered as the process parameters, which results in a 24-dimensional problem. In this example, a 3 sigma variation is considered for each process parameter.

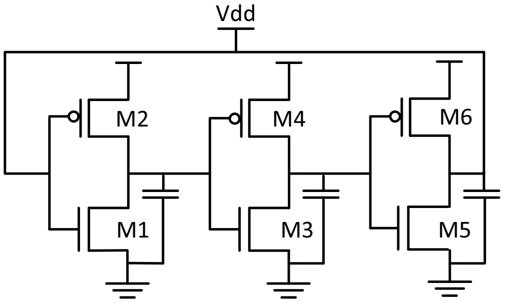

Figure 3.5: A Three-stage Ring Oscillator

Firstly, we consider 400 LHS data samples with 300 of them for training and 100 for testing. On this 24-dim problem, RReliefF is performed to reduce the dimension before constructing the frequency model. For each process parameter, the weight is evaluated and ranked as illustrated in Figure 3.6. The process parameters with negative weight are discarded and 12 parameters are kept.

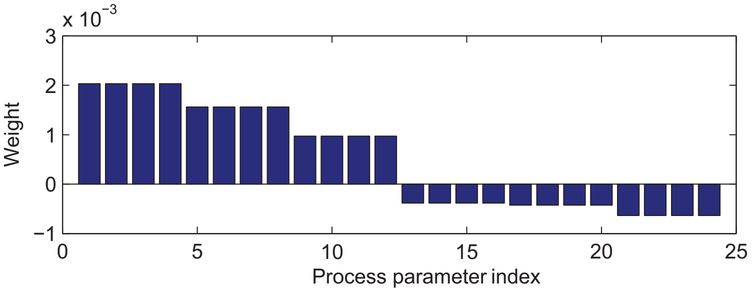

Figure 3.6: Weight of all 24 process variations for the frequency oscillation performance

We measure the oscillation frequency under the effect of the reduced set of process parameters, in order to check the accuracy of the reduction process. Figure 3.7 shows the frequency performance of 300 LHS data samples when considering the total number of process parameters (Original 24-dim) and the reduced one (Reduced 12-dim). The frequency responses are evaluated using the circuit simulator HSPICE. As it can be observed in Figure 3.7, the frequency response with the reduced set exhibits some deviation as expected. The reduction error is checked by calculating the normalized mean square error (NMSE), which is given as: $(\frac{\|freq(12-dim)-freq(24-dim)\|_2^2}{\|freq(24-dim)\|_2^2}=0.0245\%)$. The actual error is less than 0.1% which is considered excellent in practice [61].

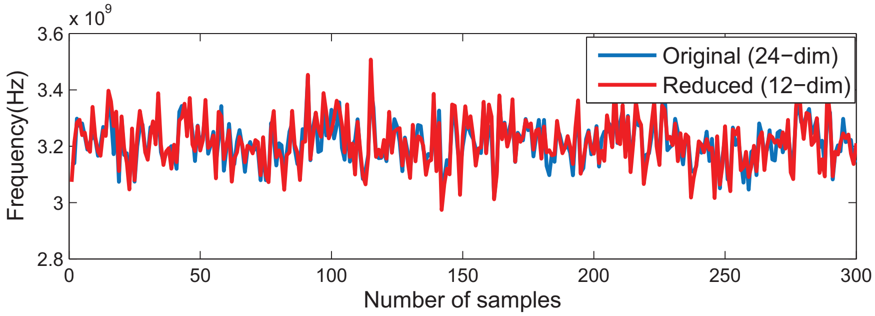

Figure 3.7: Ring Oscillator frequency responses under the original and reduced process parameters variational space

After applying the proposed adaptive LASSO scheme for surrogate modeling, we extract a frequency model of degree 3. The ability of the proposed adaptive sparse regression (ASR) modeling technique is compared to the generic sparse regression (SR) using the standard LASSO method, applied without the parameters pruning stage. The frequency of the test samples are calculated by both the constructed frequency model and HSPICE simulation. The modeling results are summarized in Table 3.1.

Table 3.1: Frequency modeling result

	ASR	SR
Fitting time (s)	85	160
Model accuracy (1-NMSE)(%)	98.65	65.61
♯ of training samples	300	
♯ of testing samples	100	

First, the ASR method appropriately selects a small subset of important monomial polynomial basis when compared to SR. Second, ASR achieves 33% better fitting accuracy than the standard LASSO. This in turn demonstrates the advantage of the weighted regression approach to consistently approximate the frequency model coefficients so that the results are not over-fitted due to the limited training set. Third, the fitting time (i.e., the cost of solving all model coefficients from the sampling points) is almost two times less than the generic SR. The fitting time reduction has been achieved thanks to the process parameters pruning.

Algorithm 3.2 computes 160 circuit simulations required to generalize and verify the frequency model accuracy. Figure 3.8 shows a graphical representation of the statistical test. The line a is the acceptance line. Similarly, the line b is the rejection line for the test. The curve intersects the line a at the observation number 160. The test is achieved at this point with a high generalized accuracy of 98.1%. At the 80^{th} and 82^{th} circuit simulation, the accuracy test has failed and the model error has been updated (i.e., generalized).

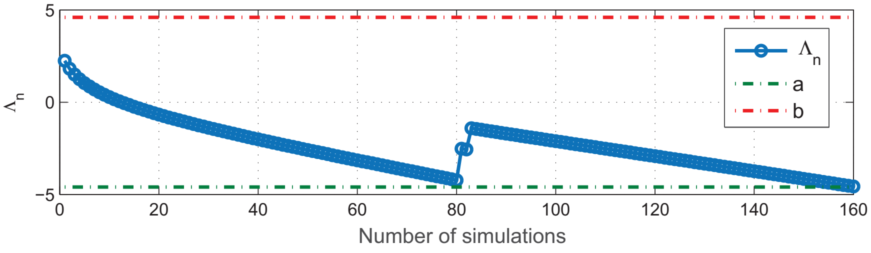

Figure 3.8: Generalization and verification of the frequency model accuracy

Table 3.2: Yield results for the ring oscillator with 24 process parameters.

Method	Total (♯) of HB sim. runs	Time Cost (h)	Yield (%)	Speed-up	Relative Error(%)
Brute-force MC	10000	4.79	73.57	1X	–
Brute-force MC	5000	2.38	71.80	2X	2.41
Quasi MC	4619	2.2125	73.57	2.16X	0.001
MC+LHS	6475	3.1015	73.88	1.54X	0.42
Our method	560	0.45	73.51	11X	0.081

In Table 3.2, we compare our results with different sampling methods including the brute-force Monte Carlo (MC), Quasi Monte Carlo (QMC) and Latin Hypercube Sampling (MC+LHS), implemented in HSPICE. Column 2 of Table 3.2 shows the number of harmonic balance (HB) circuit simulations and "Time Cost" is the time spent on simulation. The brute force Monte Carlo with 10000 is able to compute a highly accurate result of the yield rate with an estimated error $< 1\%$ at a 99% level of confidence. It is used as the golden result to assess the accuracy and efficiency of all others methods in this experiment.

For our yield estimation method, the number of HB simulations includes the number of simulations performed in the model fitting and accuracy verification phases. The 560 HB simulation runs correspond to 300 training samples, 100 testing samples and 160 samples for accuracy verification. The column "Time Cost" includes the time for all stages in the proposed methodology (i.e., the surrogate model fitting and verification, the parameter space exploration and the yield calculation). During the SMT-based parameters exploration stage, we define the full fail performance intervals as $[2.5Ghz, f_{nom} - f_{nom}2.5\%[\cup] f_{nom} + f_{nom}2.5\%, 4Ghz]$. The ring oscillator has two fail performance boundaries (P=2) and we choose a decretization step S equal to 5. In this case, $S^P = 5^2 = 25$ combinations of performance boundaries have been explored in parallel.

The SMT solver [14] has reported 2820 candidate regions. 2643 regions were confirmed by INTLAB during the solution refinement step. The regions found by the SMT solver and not confirmed during the refinement step are spurious. In this case, INTLAB refuted the existence of any solutions within the candidate regions.

On the basis of Table 3.2, it can be observed that the performance of the MC variants do not achieve significant improvement when compared to the brute-force Monte Carlo analysis engine. QMC is able to reach the MC golden result with around 2.16X speedup, while the MC+LHS method is 1.54X times faster than MC with approximately the same yield rate. Collecting extra random samples for MC+LHS does not help to converge exactly to the MC golden estimation. This observation can be explained by a bad exploration of the parameters space and a moderate uniformity properties of MC+LHS in this 24-dimensional problem.

Since the proposed method attempts to ensure an exhaustive coverage of the failure regions in the parameters space, it tends to under-estimate the yield. It explains why the predicted yield from our procedure is slightly lower than the sampling yield from MC simulations. However, the computed yield rate is almost identical to that estimated by the brute-force Monte Carlo engine with 10000 samples. Algorithm 3.3 completed the search for the failure sub-regions in $0.16h$, which is affordable and clearly demonstrates the scalability of the proposed method. In fact, the SMT problem subdivision allowed the reduction of the search space (i.e., failure performance space), and when coupled with the parallel implementation, it highly relieves the computational cost of the SMT solver.

Our method can achieve 11X speedup over the MC method while it adopts a more exhaustive approach for the yield estimation. The achieved speedup can be explained by: (1) the process parameters reduction step; (2) the employment of a surrogate model of the frequency model to replace time consuming transistor-level HB simulation; and (3) the tracking of a complete hyperrectangle in the parameters space rather than one sample point which allows a faster coverage of the failure regions.

Table 3.3: Yield results for the ring oscillator with 3 process parameters.

Method	Total (♯) of HB sim. runs	Time Cost (h)	Yield (%)	Speed-up	Relative Error(%)
Brute-force MC	10000	4.79	89.95	1X	–
Our method	380	0.1813	90.02	26.42X	0.077

In order to illustrate the capability of our method in handling multiple and distinct failure regions, we use a simplified process variational space, which only considers the threshold voltages of the n-MOS transistors $M1$, $M3$ and $M5$ as the sources of process variations. In this experiment, we apply the proposed surrogate modeling scheme without the parameters pruning stage and we formulate the SMT constraints to locate the failure regions in this 3-dimensional problem. The results are summarized in Table 3.3. As less process variables are taken into account, the time cost has significantly decreased compared with the 24-dimensional problem and the yield rate has also increased. The failure sub-regions located by our method and the fail samples of the brute-force MC engine can be clearly visualized on a 3-dimensional parameters space as shown in Figures 3.9(a) and 3.9(b), respectively. The data is projected on the three directions $(V_{thM1}, V_{thM3}, V_{thM5})$ of the 3-dimensional space, where $V_{thMi} = V_{th0Mi} + \Delta V_{thMi}, i = 1, 3, 5$.

Figure 3.9(b) shows that, similarly to the MC method, the proposed method locates two failure regions. The two regions result from the interval specification of the frequency performance metric which can be equivalently expressed as two conflicting specifications. For both methods, the frequency specification is violated for high and low threshold voltage variations of the n-MOS transistors of M_1, M_3 and

M_5. However, while the MC method randomly samples the process parameters probability distribution $pdf(p)$ towards locating the failure operation, our method directly locates three dimensional failure sub-regions in the parameters space. Also, during the SMT-based parameters space exploration, the process parameters are modeled as a set of intervals in the SMT constraints. It explains why the located failure sub-regions cover the complete parameters space in Figure 3.9(b) and differ from the failure characterization of the brute-force MC method in Figure 3.9(a). It is only at the yield calculation step that the $pdf(p)$ of the process parameters are taken into consideration to estimate the probabilistic hypervolume of each single sub-region as given in Equation 3.14.

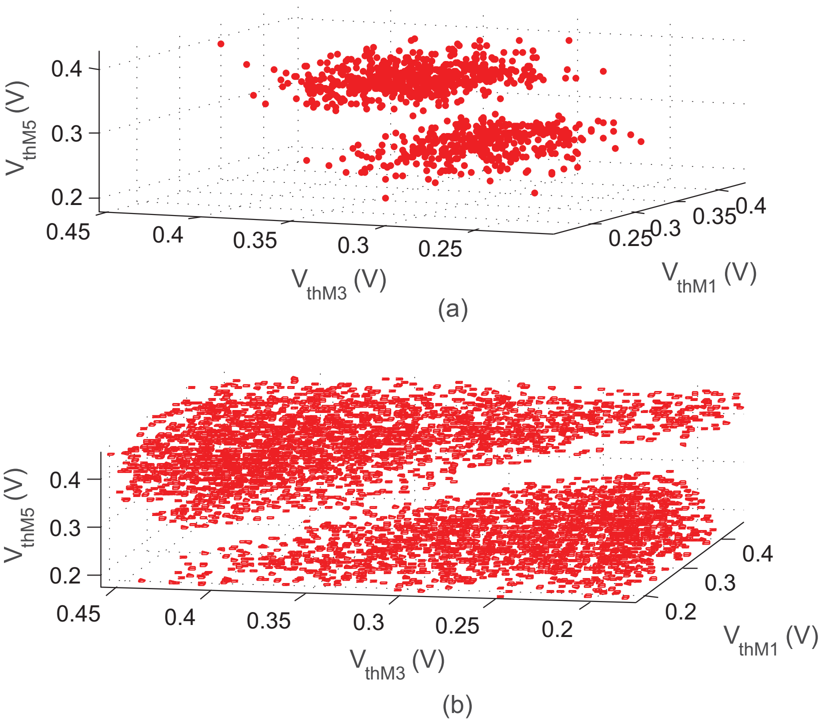

Figure 3.9: (a) Fail samples of the brute-force MC method (b) 3-dimensional failure sub-regions probed by the proposed method.

Although the proposed approach may miss some failure sub-regions due to the modeling error, the probabilistic hypervolume of the located sub-regions still can be employed to estimate the yield with 0.077% relative error when compared with the MC method. Based on this example, the ability of the proposed method in solving problems with multiple failure regions is verified.

3.2.2 6-Transistor SRAM Cell

In this section, a standard 6-T SRAM cell [62], shown in Figure 3.10, is used to validate the proposed method on a circuit with extremely high yield probability (i.e., very low failure rate (P_f)). In this example, a larger number of sigma variation (6σ) is considered. We also suppose that the brute-force MC method converges when the relative standard deviation of the failure probability ($std(P_f)/P_f$) is equal to 0.1, (i.e., 90% accuracy with 90% confidence) [63]. The SRAM cell is used to store one memory bit: the four transistors M1, M2, M3 and M4 have two stable states, i.e., either a logic 0 or 1, and the two additional access transistors M5 and M6 serve to control the access to the cell during read and write operations. All transistors lengths are set to 65nm. The width of both access transistors M5 and M6 is $0.3\mu m$. The width of the p-MOS transistors M3 and M4 is $0.2\mu m$. The width of the n-MOS transistors M1 and M2 is $0.4\mu m$.

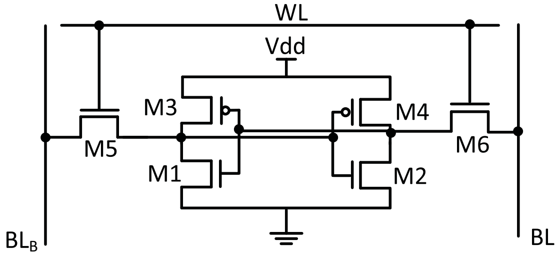

Figure 3.10: Schematic of a 6-T SRAM cell

The circuit performance is chosen as the read static noise margin (SNM) to evaluate the stability of the SRAM cell during read operation. To measure the SNM in the read operation, the word line WL is enabled and both bit lines BL and BL_B are pre-charged high. The SNM is defined as the maximum value of DC noise voltage that can be tolerated by the SRAM cell without changing the stored bit [62]. A positive value of SNM represents a stable read operation while a zero or negative value of SNM signifies that the read operation will cause the cell to lose its state, resulting in the read stability failure. We measure the SNM using a graphical technique that is based on the voltage transfer curves (VTC) characteristic of the two cell inverters. The method is explained in details in [62].

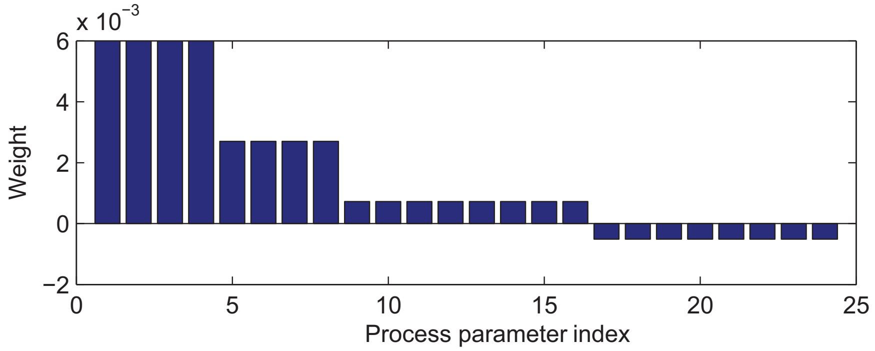

Figure 3.11: Weight of all 24 process parameters for the SNM performance

The local mismatch variables Δt_{ox}, ΔV_{th}, Δw and ΔL of each transistor are considered as the process variables, which results in 24 process parameters. On this 24 dimensional problem, RReliefF is applied to reduce the dimension before constructing the SNM performance model. For each process variation parameter, the weight is evaluated based on 300 training samples. The reduction process discarded 8 process parameters as it can be observed in Figure 3.11. Figure 3.12 plots the SNM responses simulated by HSPICE, under the effect of the full and reduced process parameters set. We evaluate the NMSE to estimate the responses deviation. The computed error is 0.5% which is low and can be considered as negligible.

We apply the adaptive LASSO scheme for modeling the SNM surrogate model.

We extract a polynomial model of degree 2 and we use 100 test samples to evaluate its accuracy. We verify and generalize the SNM model accuracy. The accuracy verification result is shown in Figure 3.13. Algorithm 3.2 computes a generalized model accuracy equal to 98.7% based on 128 simulation runs.

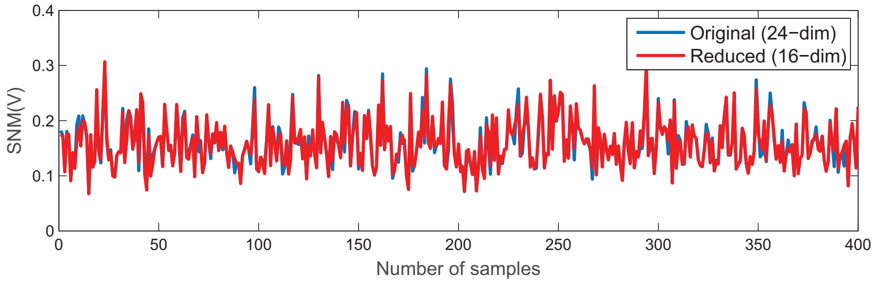

Figure 3.12: SNM responses under the original and reduced process parameters space

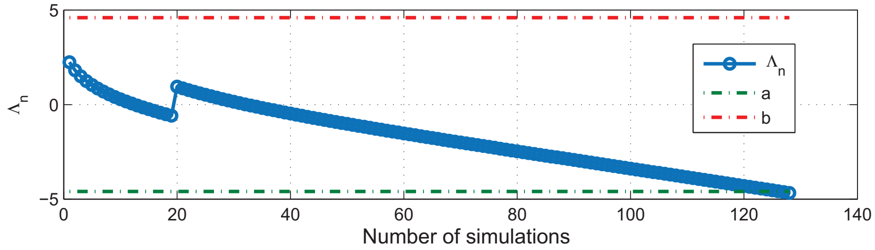

Figure 3.13: Generalization of the SNM model accuracy

The experimental results are summarized in Table 3.4. We define the full SNM fail interval as: $[-0.3V, 0V[$ and we subdivide the SMT problem into 5 sub-problems that we solve in parallel according to Algorithm 3.3. Column 2 of Table 3.4 reports the number of simulations performed in the SNM model fitting and accuracy verification phases. The column “Time Cost” shows the time for the total stages in the proposed methodology.

The MC method tries to randomly select samples to cover the entire parameters space, so it needs a huge number of samplings to achieve the target 90% level of accuracy as shown in Figure 3.14. QMC is able to reduce the number of samplings

Table 3.4: Yield results for the SRAM with 24 process parameters.

Method	Total (♯) of DC sim. runs runs	Time Cost	P_f	Speed-up	Relative Error
Brute-force MC	$4.146090e^{+6}$	19.1951 Days	$7.2357e^{-5}$	1X	-
Quasi MC	$2.093045e^{+6}$	9.6903 Days	$7.2144e^{-5}$	1.9809X	0.29%
Our method	528	0.2484 hours	$7.2468e^{-5}$	1855X	0.15%

by covering the entire space with deterministic sequences. It can be observed that the QMC method achieves around 2X speedup over the MC method with very close failure rate estimation. The method we propose in this work achieves a speedup of approximately 2000X compared with the MC method.

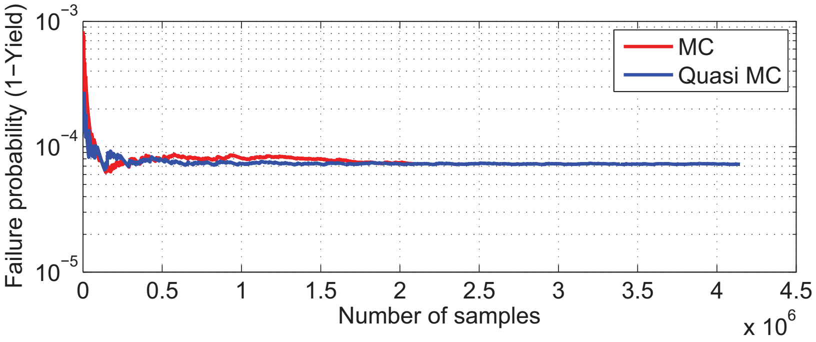

Figure 3.14: Evolution of the failure rate estimation as function of samples for the brute-force MC and the Quasi MC method

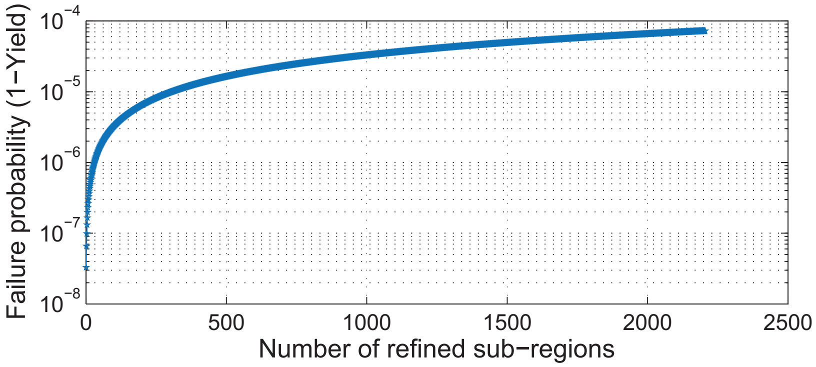

Figure 3.15: Evolution of failure rate estimation as function of tracked failure sub-regions for the proposed method

As shown in Figure 3.15, the proposed algorithm covers the failure region in the parameters space within 2205 located regions, which explains the relief in terms of computational cost. The first 400 refined regions had more contribution to the failure rate estimation in terms of probabilistic hypervolumes. The method also reaches a higher failure probability (P_f) compared to the MC method. This can be explained by the approach adopted in the proposed methodology that concentrates on the localization of only the failure regions in the parameters space. Meanwhile, the sampling methods waste a large number of samples that are far from the failure region.

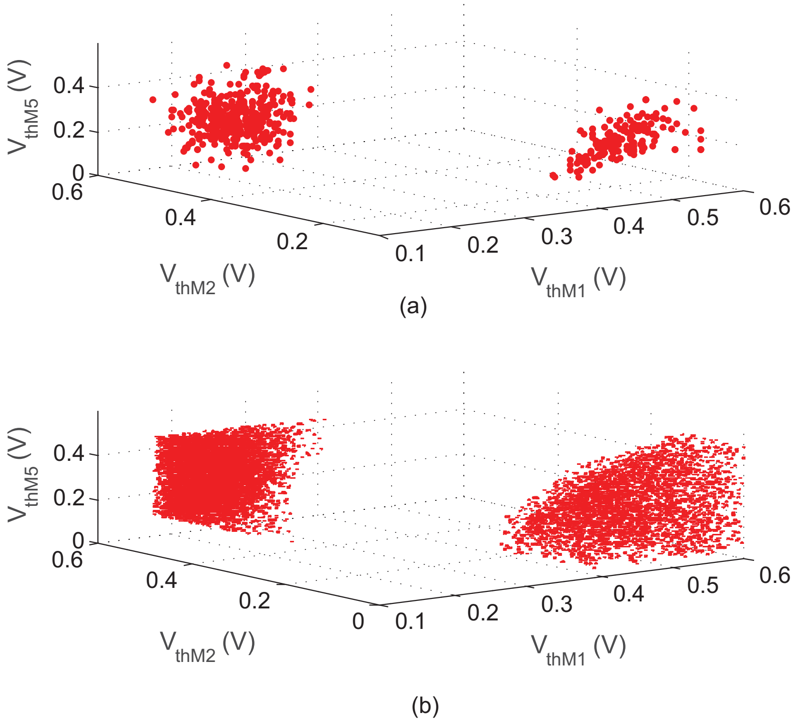

Figure 3.16: (a) Fail samples drawn from the simulation of the brute-force MC (b) Failure sub-regions located by the proposed method

Fail samples of the MC simulation result are drawn in Figure 3.16(a) which clearly shows two regions with rare failure samples. The failure occurs for asymmetrical local V_{th} variation affecting the adjacent pulling-down transistors $M1$ and $M2$. A similar localization of the failure region is reached by the proposed yield analysis scheme as it can be observed in Figure 3.16(b). In both figures, the simulation data is projected on the three directions ($V_{thM1}, V_{thM2}, V_{thM5}$) for visualization purposes. The proposed failure regions localization technique neutralizes the rare failure event issue of the SRAM circuit. Based on this example, the advantage of the proposed method in locating very rare failure regions has been demonstrated.

3.2.3 Three-stage Operational Amplifier

In this section, we will verify that the proposed yield estimation method is suitable for solving problems with multiple performances specifications as well as high dimensional parameters space. We consider a three-stage amplifier (op-amp) [64] as shown in Figure 4.5. The width to length ratio of all unlabeled n-MOS and p-MOS is $\frac{10}{2}$ and $\frac{22}{2}$, respectively.

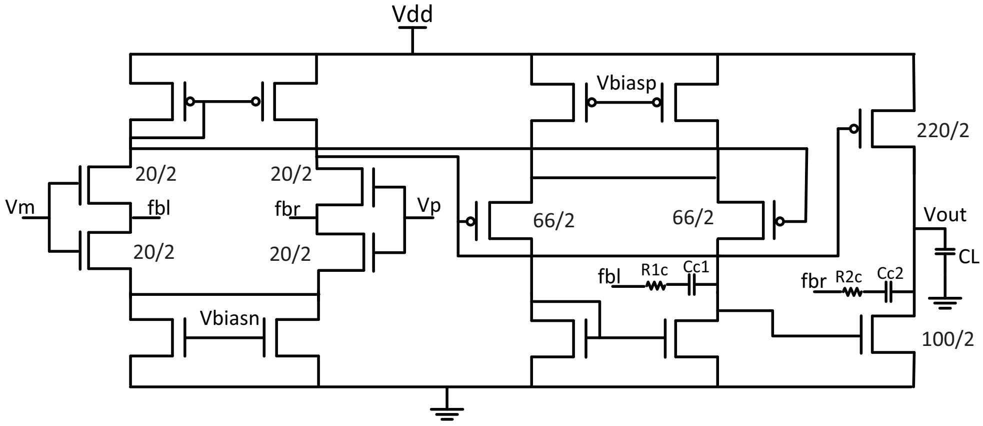

Figure 3.17: A Three-stage operational amplifier

We select Δt_{ox}, ΔV_{th}, Δw and ΔL as the process variables. The local mismatch in each transistor pair is considered. It leads to a total of 56 process parameters. In this example, a 3 sigma variation is considered for each process parameter. The performance of the circuit is characterized by many properties, such as voltage gain

(Av), phase margin (PM), the gainbandwidth (GBW) and the DC offset voltage ($DCOffset$). The op-amp is designed to satisfy the list of specifications shown in Table 3.5.

Table 3.5: The set of specifications for the three-stage op-amp

Perf metrics	Simulation	Specification
$Av(dB)$	AC	≥ 40
$GBW(MHz)$	AC	≥ 80
$DCOffset(mV)$	DC	≤ 50
$PM(^{\circ})$	AC	≥ 60

Firstly, 300 initial LHS simulations are used to build a surrogate model of for all properties. 200 of them are employed for model training and 100 for subsequent model testing. Each property is measured using a specific type of simulation. Note that even though we analyze and model each performance metric individually, these performance metrics are not necessarily independent as they are sharing the transistor-level simulations of the pre-sampling stage. In fact, by evaluating all performance metrics for each individual sample drawn from the process parameters space, we substantially reduce the total number of simulation runs and, hence, the computational cost.

Table 3.6: Result of the process parameters reduction stage

Perf metrics	Reduced Set (♯)	Reduction Error
$Av(dB)$	32	0.79%
$GBW(MHz)$	24	0.95%
$DCOffset(mV)$	40	0.85%
$PM(^{\circ})$	44	0.95%

On this 56-dim problem, RReliefF is performed to reduce the dimension of the process parameters. The experimental results of the reduction process are summarized in Table 3.6. It can be observed that in this example the dimension of the original set of process parameters for each performance metric did not largely decrease. This can be explained by the consideration of multiple performance metrics that depend on most of the process variables. Furthermore, the accuracy of the circuit response under the reduced set of process parameters is maintained.

Table 3.7: Surrogate models degree and accuracy (1-NMSE)%

Perf metrics	Degree	Model Accuracy	Gen-Accuracy
Av	3	98.0%	97.8%
GBW	1	98.1%	98.05%
DCOffset	1	98.8%	98.3%
PM	3	98.7%	98.2%

We evaluate the accuracy of the models trained using the adaptive LASSO scheme. We report the final degree of the approximations and the models accuracies in Table 3.7. In the "Degree" column of Table 3.7, we see that for some properties, we are able to construct polynomial models with a degree lower than the limit $D = 3$. The accuracy generalization step statistically verifies the op-amp properties model with respect to the reduced set of process parameters. In the column "Gen-Accuracy", we report the result of the accuracy generalization stage. We can find that the accuracy is more than 97% for all models.

We apply the brute-force MC, Quasi MC and MC+LHS to estimate the yield of the op-amp. The results are reported in Table 3.8. The brute-force MC method is run with a target accuracy of 99% and a confidence level of 95%. For the sampling methods, "Time Cost" is the circuit simulation time and "Sim(♯)" refers to the number of samples. The column "Sim(♯)" in our method includes the number of circuit simulations performed in the surrogate model fitting and accuracy verification phases. "Sim Time" shows the total circuit simulation time and "Fitting/Verif Time" indicates the time spent in the model fitting and verification stages excluding the circuit simulation time. "Time" is the time spent in the parameter space exploration and the yield calculation. Finally, “Time Cost” is the total computational time.

Table 3.8: Yield results for the Op-amp with 56 process parameters

Perf metrics	Brute-force MC	MC+LHS	Quasi MC	Our method			
	Sim($\sharp$)	Sim($\sharp$)	Sim($\sharp$)	Sim($\sharp$)	Sim Time	Fitting/ Verif Time	Time
Av	8740	6650	6420	300/135	0.28h	96s/3s	0.26h
GBW	8740	6650	6420	300/119		5s/4s	
DCOffset	8740	6650	6420	300/69		6s/3s	
PM	8740	6650	6420	300/201		91s/6s	
Time Cost	2.97h	2.26h	2.18h	0.60h			
Speedup	1X	1.32X	1.37X	5X			
Yield (%)	81.61	79.60	81.6	80.53			
Relative Error	-	2.46%	1.24%	1.32%			

As in the previous experiments, we observe that the predicted yield from our approach closely matches the yield estimation of the MC method. Our method requires fewer simulations and finishes faster with a speedup of almost 5X. This application shows again the benefits of a model building approach rather than direct yield estimation from a circuit simulator. Also, the column “Fitting/Verif Time” in our method shows that even though the reduction result was not very significant, the proposed adaptive sparse regression algorithm still renders the fitting time quite affordable. This result further demonstrates the scalability of the proposed technique to handle larger problems. The regression time of the model performance with a degree lower than the degree limit (i.e., *GBW* and *DCOffset*) is significantly smaller. In fact, the major cost in regression lies in the computation of the LASSO coefficients. The former can be easily parallelized, leading to further performance improvements.

3.2.4 Parameters Discussion

The surrogate-based yield estimation requires the setting of several parameters. These parameters can have a large effect of the accuracy of the computed yield rate. In this section, the key parameters in the proposed method are discussed. The parameters values that in practice lead to reliable results are reported.

Parameter R_{th} in Algorithm 3.1

We construct the frequency model of the Ring Oscillator example in Section 3.2.1 with different R_{th} from 9.10^{-2} to 1.10^{-2}. Figure 3.18 shows the error of the yield rate with respect to the model accuracy defined as $(1 - R_{th})\%$. The error of the yield is computed relatively to the yield result of 10000 MC simulations run. We can find that when the accuracy is smaller than 97%, the relative error resulting primarily from the fitting error of the frequency model increases signicantly. To ensure the viability of the proposed method, we must ensure that the accuracy is high enough at the modeling stage. So, in practice, the value of R_{th} should be selected from 3.10^{-2} to 1.10^{-2}.

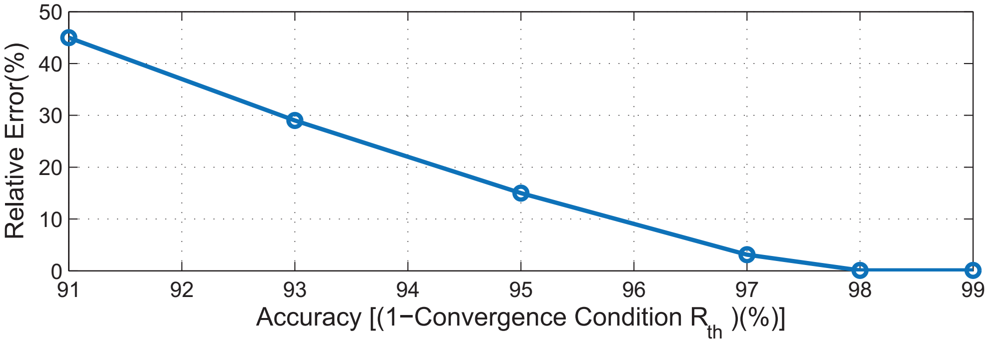

Figure 3.18: Relative error with respect to R_{th}

Parameters (α, β, φ) in Algorithm 3.2

We applied Algorithm 3.2 to verify and generalize the frequency model accuracy $freq(\tilde{p})$ of the Ring Oscillator example in Section 3.2.1. We checked the property:

$$\forall p, \tilde{p} \in P \quad Pr((err(f(p), freq(\tilde{p})) \leq \varepsilon) \geq \ \varphi \tag{3.15}$$

where $\varepsilon = 0.0135$ (i.e., 98.65% accuracy). We applied the algorithm for different values of φ and equal error rates (α, β). We used an indifference region $[\varphi - \delta, \varphi + \delta]$ where $\delta = 0.001$. The results are summarized in Table 3.9. Increasing φ and decreasing (α, β) requires a larger number of simulations, leading to a model verification with better statistical guarantee. The model accuracy has been verified and generalized to 0.019 (i.e., 98.1% accuracy). In practice, we find that $\varphi = 0.95$ and $\alpha = \beta = 0.01$ often provide a good trade-off between statistical guarantee and computational cost.

Table 3.9: Run length for common values of φ and (α, β)

$\alpha(= \beta)$	0.02	0.01	10^{-3}
φ=0.9	37	44	65
0.95	75	160	214
0.99	683	762	1015

Tolerance margin in failure performance bounds of Equation 3.11

If the circuit specification includes a performance metric f that should be greater than a limit f_{limit} (i.e., $f > f_{limit}$), then the failure performance region is defined as $f \in [f^l, f_{limit}]$. If the value f^l is over-approximated (i.e., it is below the value that can be reached in reality), it will not affect the result of the yield estimation and it will slightly affect the computation time. In fact, the SMT solver rapidly discards parts from the search space that contains no solutions. However, if it is under-approximated (i.e., it is greater than the value that can be reached in reality), it will prevent the SMT solver from locating failure regions in the parameters space and affect the final yield estimation. In practice, we firstly set $f_l = f_{min} - \Delta f$, where $\Delta f = |f^{min} - f_{nom}|$, f_{nom} is the nominal value of f and f^{min} is the minimum value of f discovered during the initial pre-sampling and circuit simulation step. If the SMT solver discovers failure regions in the parameters space with performance values f in the neighborhood of f_l, that is $f \in [f_l, f_l + 3\varepsilon]$, where ε is the model error, then f_l should be further decreased by Δf. Otherwise, the user can be highly assured that the failure performance bounds have been conservatively characterized.

3.3 Summary

This chapter presented a methodology for analog circuits yield analysis. Different techniques such as parameters pruning, adaptive sparse regression and sequential probability ratio test were used to build performance models and verify their accuracy. We then employed an SMT solving technique and interval arithmetic to exhaustively probe the parameters space and to locate the failure regions of the circuit operation. The yield is calculated based on the probabilistic volume of the located failure regions. Compared with existing methods, the surrogate-based yield estimation method tried to handle yield problems with: (1) many process parameters; (2) multiple and distinct failure regions; (3) multiple performances specification; and (4) extremely high yield rate. The experimental results on several analog circuits show that the method is reliable while leading to a simulation speedup when compared to the brute-force MC.

We enhanced the run-time and scalability of SMT solving techniques by adopting multiple strategies including: (1) reduction of the SMT problem variables; (2) building low complex performances models; (3) reduction of the SMT problem search space; and (4) adjustment of the SMT solver resolution and solution refinement. Furthermore, the computational cost of the proposed surrogate modeling algorithm has been enhanced by reducing the number of process parameters and avoiding a high polynomial degree. The run time of the adaptive sparse regression may largely increase if more aggressive process variation and a larger number of process parameters are considered. Efficient and more advanced modeling and parallelization techniques may tackle this limitation.

The proposed methodology can be integrated with an optimization technique which aims at finding a circuit design that has a maximum yield, considering the performance specifications and the manufacturing variation. The efficiency of the optimization process in terms of computational cost and search space coverage is critical. In the next chapter, we present an optimization process which can ensure a good coverage of the feasible design space while minimizing the computational cost.

Chapter 4

A Two-Phase Yield Optimization Method

This chapter presents a novel approach for improving analog yield optimization. The optimization is performed in two steps. A parallelized global optimization phase uses a modified Lipschitizian optimization method to locate the basin of convergence of the optimum solution. The search ensures that potentially optimal regions of the design space are not overlooked. Once a good approximation of the global optimum is located, it is exploited by a local optimization phase. The local search uses the located near optimal solution as a starting point. Also, the local optimization phase is integrated to remedy to the limitation of the Lipschitizian method by accelerating its convergence speed. The method builds interpolating models using linear combinations of Radial Basis Functions (RBF) that approximates locally the objective function and conducts a local refinement. Its efficiency is further elevated by the reuse of existing simulation data of the global search phase. We demonstrate the advantages of the proposed methodology on a folded cascode amplifier, a two-stage operational amplifier and a three-stage operational amplifier. We optimize the yield of these circuits under the effect of device mismatch and compare our method with stochastic search optimizations in terms of solution optimality and run time.

4.1 Preliminaries

Before presenting the proposed methodology, we briefly explain our main objective and define terms that will be used in the rest of the chapter.

4.1.1 Problem Definition

The problem of finding the design point x^* that maximizes a yield function g, can be formulated as a nonlinear optimization problem with bound constraints [6], as given in Equation 4.1.

$$\begin{aligned} \mathbf{x}^* &= \max_{\mathbf{x}\in\mathbf{D_0}} \mathbf{g}(\mathbf{x}) = \min_{\mathbf{x}\in\mathbf{D_0}} \mathbf{f}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) &= \mathbf{E}\{\delta(\mathbf{x},\mathbf{p})|\mathbf{pdf}(\mathbf{p})\} \end{aligned} \tag{4.1}$$

where $x \in R^n$ is the vector of design variables, which can be composed of transistor widths and lengths, bias voltages and currents, etc. Each design variable x_i is limited in a range $[x_i^l, x_i^u]$. $D_0 = \{x \in R^n, x^l \leq x \leq x^u\}$ is an n-dimensional Euclidean search space (also called a *hyperrectangle*), $g : D_0 \longrightarrow R_+$ is a positive real-valued yield function and $f = -g$. We assume that f is smooth and continuous over D_0. p is a vector of continuous random variable modeling process parameters variations, e.g., gate length ΔL and oxide thickness Δt_{ox}. $\vec{p}$ is the joint probability density function of p. E denotes the expectation value. $\delta(x, p) = 1$ if the design point x meets all specifications under process fluctuation. Otherwise, $\delta(x,p) = 0$.

4.1.2 Lipschitizian Optimization

In our yield optimization method, we propose to use Lipschitizian optimization to find a design point near the global optimum of the problem in Equation 4.1. Let f be Lipschitz continuous on $D_0 = [a,b]$, with constant K. Lipschitz optimization employs the Lipschitz property to construct an iterative algorithm that seeks the minimum of f [65]. In fact, for any $x \in [a,b]$, f satisfies [65]:

$$\begin{aligned} f(x) &\geq f(a) - K(x-a) \\ f(x) &\geq f(b) + K(x-b) \end{aligned} \tag{4.2}$$

The two inequalities in Equation 4.2 form a V-shape below f, as shown in Figure 4.1(1). The point of intersection for the two lines provides the first estimate C_1 of the lower bound of f. The method performs the same operation on $[a, x_1]$ and $[x_1, b]$ and iteratively continues dividing the interval with the smallest lower bound (Figures 4.1(2) and 4.1(3)). The V-shape of all intervals form a piecewise linear function $\hat{f}$ that approximates f, where, $\hat{f}(x) \leq f(x)$, $\forall x \in [a,b]$ (Figure 4.1(4)). The process continues until the difference between the best function value, $f(\overline{x^*})$, and the value of the smallest lower bound found after n iterations, $\min_{i\in[1,n]} C_i$, is smaller than or equal to a precision parameter $\theta > 0$.

Lipschitz optimization is globally θ-convergent [65]. In fact, it returns in a finite number of iterations $\overline{x^*}$ that satisfy:

$$f(\overline{x^*}) \leq \min_{i\in[1,n]} C_i + \theta \leq f(x^*) + \theta,\ x^* = \min_{x\in[a,b]} f(x) \tag{4.3}$$

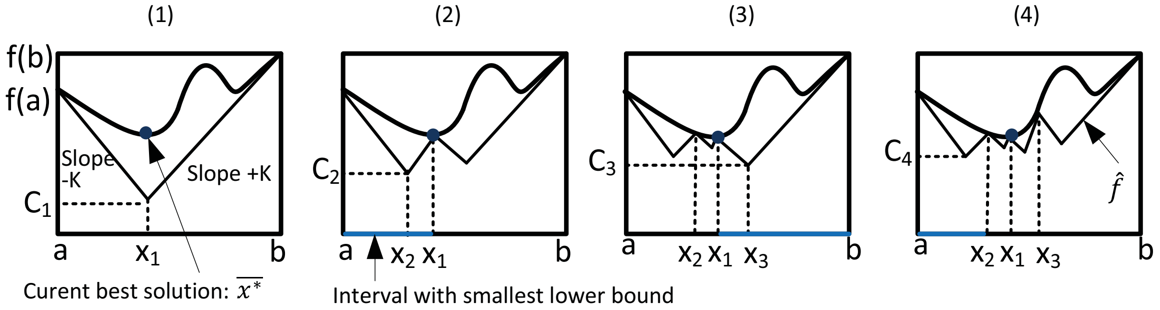
(1)
(2)
(3)
(4)
f(b)
f(a)
Slope -K
Slope +K
C1
C2
C3
C4
a
x1
x2
x3
b
Curent best solution: $\overline{x^*}$
Interval with smallest lower bound
$\hat{f}$

Figure 4.1: Univariate Lipschitz optimization iterations

The method usually needs a few function evaluations to find the area near the global optimal point. However, it requires knowledge of the Lipschitz constant and is computationally complex in higher dimensions. In this chapter, we will mainly focus on this optimization technique and develop a novel algorithm to make it tractable and computationally efficient for analog yield optimization.

4.2 Yield Optimization Methodology

The methodology in Figure 4.2 details our proposed yield optimization approach. The optimization takes as input a continuous set of interval-valued sizing solutions that characterizes the feasible design space D_0. The feasible design space is computed using the nominal circuit sizing methodology proposed in Chapter 2. The focus of this chapter is to find the most robust design $x^* \in D_0$ that maximizes the yield, where x^* is a vector of design variables, which can be composed of transistor widths, bias voltages and currents, etc. At each iteration, the global and local yield optimization phases require the yield computation of some selected design points. Various yield estimation techniques can be employed (e.g., the surrogate-based method proposed in Chapter 3, the Monte Carlo (MC) method and its variants, etc.).

The proposed yield optimization strategy is composed of a global and a local

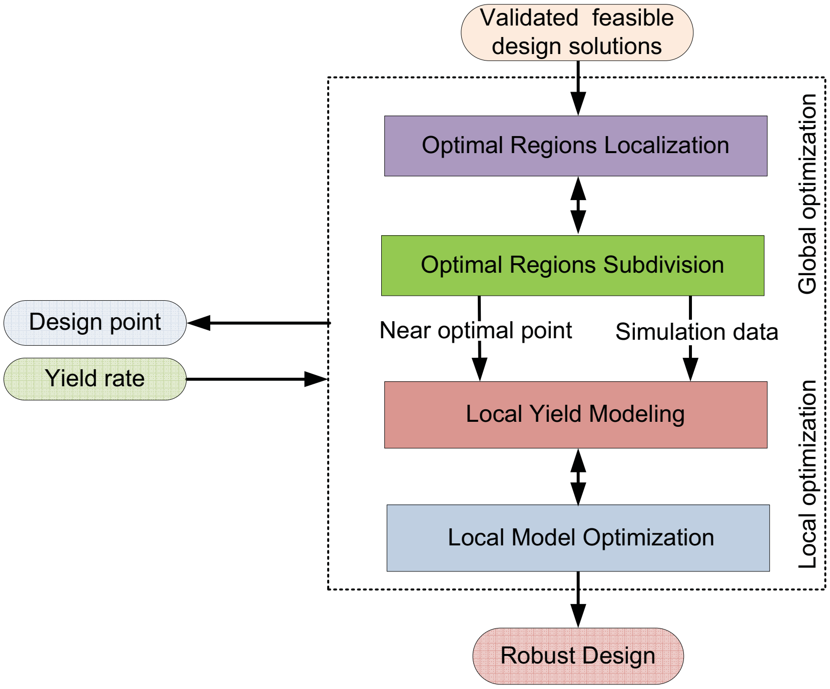

Figure 4.2: Yield optimization methodology

optimization phases. The global optimization step determines a design point close to the optimal solution that is used as a starting point by a local optimization phase. The global search uses a modified Lipschitizian algorithm to reach the area near the global optimum. The process iteratively locates and partitions potentially optimal subregions of the feasible design space D_0. The potentially optimal subregions are the largest subregions with the best yield rate at their centers. In order to ensure the computational efficiency of this stage, the global optimization stops the search when the subregion with the highest yield rate is sufficiently small. The stopping criteria trades off between the computational cost and the solution optimality. Besides, the search is subdivided into a number of subproblems that are run in parallel.

The local search mechanism is used to rapidly reach the optimal design point starting from the best solution computed by the global search. To do so, it iteratively constructs local models (i.e., around a current iterate) of the yield function using linear radial basis function (RBF). The method employs previously evaluated points (i.e., the global optimization simulation data) to accelerate the modeling stage. If the number of available points is not enough to uniquely define the models, new data points (i.e., design points and their corresponding yield rate) in the neighborhood of of the current iterate are generated. An approximated solution is computed by locally optimizing the yield model. The process is repeated until the yield model gradient is sufficiently small.

4.2.1 Parallel Global Optimization

The aim of the global search is to locate a design point $\overline{x^*}$ near the optimal solution x^* of the problem in Equation 4.1. The approach is summarized in Algorithm 4.1.

Alg. 4.1. Parallel global optimization

```
Require: $D_0$: Design search space, $f$: Objective function
$D_0^{1\rightarrow S} \leftarrow$ Divide($D_0$)
for all $ind = 1 \rightarrow S$ do in parallel
    $\overline{D_0^{ind}} \leftarrow$ normalize($D_0^{ind}$)
    Initialize: $\overline{x^*_{ind}} \leftarrow$ center($D_0^{ind}$), $f_{ind}^{min} \leftarrow f(\overline{x^*_{ind}})$, $c_0 \leftarrow \overline{x^*_{ind}}$, $f(c_0) \leftarrow f_{ind}^{min}$, $\Gamma_{ind} \leftarrow$
    $\{c_0, f(c_0)\}$
    while stopping criteria is unsatisfied do
        Identify $S$: all potential optimal hyperrectangles $H_j$
        for all $H_j \in S$ do
            Identify $M$: the dimensions with max. side length $d$,
            Evaluate in parallel $f(c_j \pm \alpha e_m)$, $m \in M$, $\alpha = d/3$
            $\Gamma_{ind} \leftarrow \Gamma_{ind} \cup \{c_j \pm \alpha e_m, f(c_j \pm \alpha e_m)\}$
            Update $\overline{x^*_{ind}}$, $f_{ind}^{min}$
            Evaluate $w_m$ and divide $H_j$ according to $w_m$
        end for
    end while
end for
return $\overline{x^*}$=min$_{\overline{x^*_{ind}}, f_{ind}^{min}}(f_{ind}^{min})$, $\Gamma = \bigcup_{ind=1}^{S} \Gamma_{ind}$
```

Algorithm 4.1 is based on the the DIRECT method [66] that is a variant of Lipschitzian optimization. Hence, it is effective in finding the basin of convergence. Also, it eliminates the need to specify a Lipschitz constant. Instead, it uses all possible values to determine if a region of the search domain D_0 should be broken into sub-regions and explored. Also, it can operate in high-dimensional space as it uses a partitioning strategy of the search spaces into hyperrectangles that requires the sampling of their center points only.

In order to decrease the optimization running time and to conduct a refined exploration of the search space, we start by subdividing the yield optimization process into S subproblems that we invoke simultaneously (Line 1). Each subproblem is limited to a sub-region of D_0 that is transformed into the unit hypercube (Line 3). The near optimal point is initialized by sampling the center of the search space (Line 4). Then, the set of potentially optimal hyperrectangles S is identified (Line 6). A hyperrectangle H_j is said to be potentially optimal if there exists a rate of change constant $\overline{K} > 0$ such that:

$$\begin{aligned} f(c_j) - \overline{K}d_j &\leq f(c_i) - \overline{K}d_i, \forall i \in I \\ f(c_j) - \overline{K}d_j &\leq f_{ind}^{min} - \gamma|f_{ind}^{min}| \end{aligned} \tag{4.4}$$

where I is the set of all indices of all hypererctangles, c_j is the center of H_j and d_j is the size of H_j defined as the distance from the center to the vertices of H_j [66]. The mathematical formula of d_j can be found in [66]. f_{ind}^{min} is the current best function value. The first inequality in Equation 4.4 expresses the decision to choose the hyperrectangle which promises the best improvement (i.e., decrease) in the objective function. It also ensures that as soon as a larger hyperrectangle with a lower function value at its center exists, the algorithm switches the search to that more promising (i.e., potential) region. Also, it is not required to specify the value of $\overline{K}$. Instead, the algorithm searches for any possible strictly positive value. The parameter γ in the second inequality guarantees that there is a sufficient decrease in the objective function. Once H_j is identified as potentially optimal, it is divided into smaller hyperrectangles (Lines 7 to 13). The divisions are performed only along its longest sides. It starts by determining the set M of all dimensions of maximal length (Line 8). Then, the function f is evaluated in parallel at $c_j \pm \alpha e_m$, where α is one-third the maximum side-length, and e_m, $m \in M$ is the m^{th} unit vector (i.e., a vector with a one in the m^{th} position and zeros elsewhere) (Line 9). The first division is performed perpendicular to the side with the lowest w_m, where:

$$w_m \quad = \quad min\{f(c_j + \alpha e_m), f(c_j - \alpha e_m)\},\ m \in M \tag{4.5}$$

The new hyperrectangle that has center c_j is divided perpendicular to the direction of the second lowest w_m. The process is repeated until H_j is divided in all directions $m \in M$. The subdivision ensures that previous function evaluations are at the center of the new hyperrectangles (Figure 4.3).

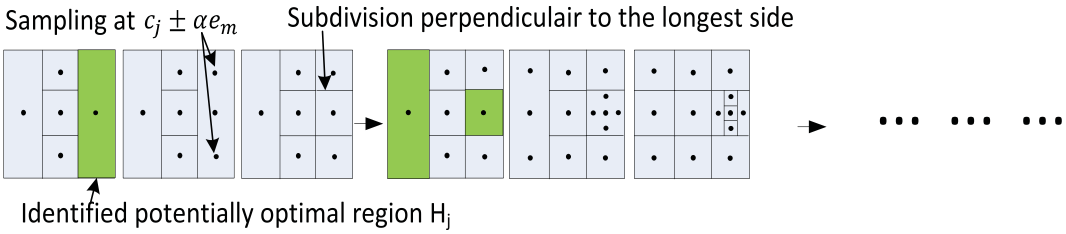
Sampling at $c_j \pm \alpha e_m$
Subdivision perpendicular to the longest side
Identified potentially optimal region H_j

Figure 4.3: Two global optimization iterations

The global convergence of Algorithm 4.1 may come at the cost of a slow optimization at the final phase. In fact, the complexity escalates as the number of subdivided subregions increases. We overcome this limitation by stopping the search when the hyperrectangle with the lowest objective function is sufficiently small. At this stage, the subdivisions become clustered near the global solution and the algorithm enters a refinement stage. The *stopping criteria* is given as $d_j < \sigma d_0$. It stops the search when the size (i.e., d_j) of the hyperrectangle H_j with the best objective function at its center c_j reaches a certain percentage of the original unit hypercube size d_0. $0 < \sigma < 1$ is adjusted for a trade-off between computational cost and the solution optimality. It should ensure that no region is omitted and minimizes the risk of a premature termination.

The outputs of of Algorithm 4.1 are the best solution $\overline{x^*}$ reached by the subproblems and the simulation data Γ (Line 16), where Γ is composed of all sampled center points and their function evaluations.

4.2.2 Local Optimization

After a design point $\overline{x^*}$ in the basin of convergence is identified, a local search is invoked to speed up the convergence. The local search iteratively builds and optimizes a linear and non expensive RBF model of the objective function within a neighborhood of a current iterate. At each iteration, it solves the subproblem of the form [67]:

$$\begin{array}{rcl} \min \quad m_k(x_k+s), & & x_k+s \in B_k \\ B_k & = & \{x_k+s, s \in R^n : \|s\|_2 \leq \triangle_k\} \\ m_k(x_k+s) & = & \displaystyle\sum_{i=1}^{|\Psi|} \lambda_i \phi(\|s-y_i\|_2) + p(s) \\ f(y_i) & = & m_k(y_i), \quad \forall y_i \in \Psi \end{array} \tag{4.6}$$

where x_k is the current state, B_k is the so called trust region for an implied (center, radius) pair (x_k, $\triangle_k > 0$) and $\|.\|_2$ is the l_2 norm. The model m_k approximates f

within a neighborhood of the current trust region B_k. It is a linear combination of RBFs (Line 3 in Equation 4.6). $\phi : R_+ \rightarrow R$ is a univariate RBF. λ_i are the linear model coefficients, which are determined by requiring that the model m_k interpolates the function f at a set of linearly independent data points $\Psi = \{y_i, f(y_i)\}$ (Line 4 in Equation 4.6) at which the values of f are known, including the current iterate x_k. The interpolation results in a system of linear equations [67]. $p(s)$ is a low order polynomial tail. $|\Psi|$ is the cardinality of Ψ. Algorithm 4.2 illustrates the method at each iteration.

Alg. 4.2. Local optimization

```
Require: Γ: Available simulation data points, $x_0 = \overline{x^*}$: Starting point, $f$: Objective
   function
1: while $\|\nabla m_k(x_k)\| \geq \varepsilon$ do
2:    Select $\Psi \in \Gamma$ in the neighborhood of $B_k$.
3:    Build $m_k$ interpolating $f$ at $\Psi$
4:    minimize $m_k$ within $B_k$ and compte $s_k$
5:    Evaluate $f(x_k + s_k)$ and update $\Gamma$
6:    Compute $\rho_k$ and adjust $B_k$
7: end while
8: return $x^*$: optimal solution
```

The current iterate x_k is usually surrounded by several neighbored points which have been evaluated previously in Algorithm 4.1. These simulation data points are reused to accelerate the local optimization phase. That is, at each iteration, the algorithm selects a set of data points $\Psi \in \Gamma$ within a neighborhood of the trust region B_k (Line 2). If the neighboring points are not enough for linear interpolation (i.e., they do not guarantee the non singularity of the interpolation problem and the uniqueness of the model unknown coefficients λ_i), new points in the neighborhood of x_k are properly generated [67]. Then, the model m_k that interpolates f is built (Line 3) and the unknown model coefficients λ_i are determined. The model m_k is assumed to approximate the objective function sufficiently well in the current trust region B_k.

The approximated solution s_k (i.e., the step) is computed by optimizing m_k over the trust region B_k (Line 4). The yield is evaluated at $x_k + s_k$ and the set Γ is updated (Line 5). In fact, any evaluated design point is saved in Γ, which allows the algorithm to gain additional insight into the function in the next iterations.

The pair $(x_k, \triangle_k)$ of the trust region B_k is adjusted according to the ratio of the achieved versus the predicted improvement (i.e., decrease of the objective function f), $\rho_k = \frac{f(x_k)-f(x_k+s_k)}{m_k(x_k)-m_k(x_k+s_k)}$ (Line 6). If ρ_k is sufficiently positive, then the iteration is successful; the next iteration point $x_{k+1} = x_k + s_k$ will be taken and the trust-region radius $\triangle_k$ is enlarged. If ρ_k is not sufficiently positive, then the iteration was not successful; the current x_k will be kept and the trust-region radius is reduced. The process is repeated until the model gradient $\|\nabla m_k(x_k)\|$ is smaller than a threshold parameter ε. That is, the sequence of x_k converges to a stationary point. The convergence criteria proof can be found in [67].

4.3 Applications

In this section, we present the results of the application of our yield optimization technique on three standard amplifier circuits. In the experiments, the circuits are designed in a commercial TSMC 65 nm process and simulated in HSPICE with BSIM4 transistor models. The local mismatch variables are considered as the process parameters including the oxide thickness $\triangle t_{ox}$, threshold voltage under zero bias $\triangle V_{th}$, channel width $\triangle w$ and channel length $\triangle L$. We use the TSMC 65 nm transistor mismatch model with $Vdd = 1V$ and standard threshold voltage. Each process parameter follows a truncated normal distribution. We compare our method with stochastic search algorithms including: Genetic Algorithm (GA), Differential Evolution (DE) and Simulated Annealing (SA). These optimization routines are linked to the circuit simulator HSPICE to evaluate the yield of a design point. The yield evaluation uses MC analysis with the LHS technique and 2000 samples. This number provides a relatively accurate result when compared to the result of 70000 simulations run. The application of our yield optimization methodology on the the first two amplifier circuits uses HSPICE for yield evaluation. For the third circuit, it uses the surrogate-based yield estimation method presented in Chapter 3. In Algorithm 4.1, we set $\gamma = 10^{-3}$ and we subdivide the optimization into $S = 4$ subproblems. Algorithm 4.2 uses sequential quadratic programming (SQP) and cubic RBF models.

4.3.1 Folded Cascode Amplifier

We consider a folded cascode amplifier [6] circuit as shown in Figure 4.4. Table 4.1 provides the specifications for the gain Av, gainbandwidth GBW, power P_{DC}, slew rate SR and DC offset voltage $DCOffset$.

Table 4.1: Set of specifications for the folded cascode amplifier

Perf metrics	Spec
$Av(dB)$	≥ 20
$GBW(MHz)$	≥ 5
$P_{DC}(mW)$	≤ 0.6
$SR(V/\mu s)$	≥ 30
$DCOffset(mV)$	≤ 40

The length of all transistors is fixed to 130nm. After applying the symmetry constraints, the number of independent design variables is 9. The nominal sizing solutions that correspond to the optimization search space D_0 are reported in Table 4.2.

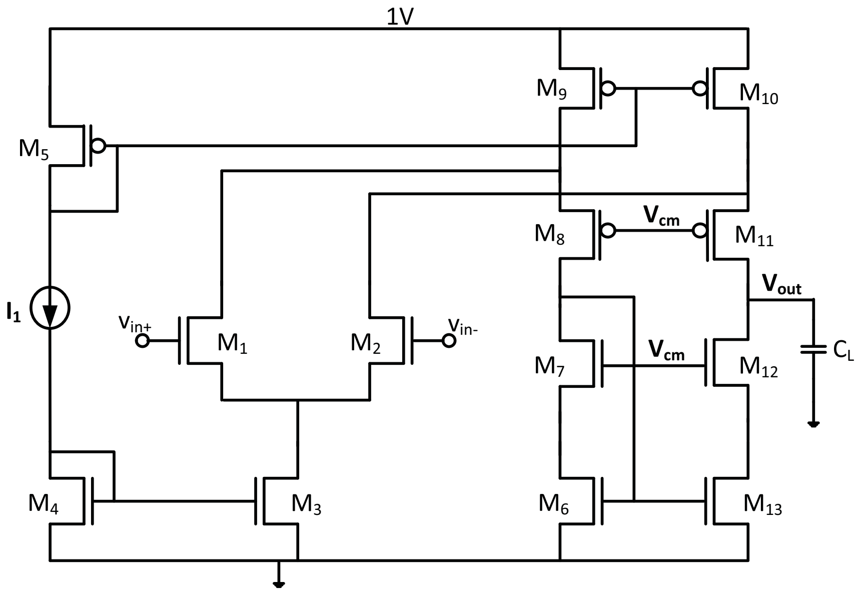

Figure 4.4: Fully differential folded cascode amplifier

The local mismatch variables of each transistor pair are considered, which results in 24 process parameters. The results of the proposed yield optimization approach are reported in Table 4.3. PGOpt refers to the parallel global optimization (i.e., Algorithm 4.1) and LOpt refers to the local optimization (i.e., Algorithm 4.2). The number of yield evaluations and the yield values reached by each phase are reported in Columns 2 and 3, respectively.

Table 4.2: Design variables ranges of the folded cascode amplifier

Design variables	Ranges
$w_1 = w_2(\mu m)$	[2.63, 8.9]
$w_9 = w_{10}(\mu m)$	[3.04, 5.04]
$w_8 = w_{11}(\mu m)$	[4.85, 8.95]
$w_7 = w_{12}(\mu m)$	[0.61, 2.64]
$w_6 = w_{13}(\mu m)$	[0.62, 2.61]
$w_3 = w_4(\mu m)$	[4.2, 5.6]
$w_5(\mu m)$	[6.1, 7.8]
$I_1(\mu A)$	[252, 268]
$V_{cm}(V)$	[0.450, 0.451]

We also perform the optimization with different stopping criteria parameter values σ of the global optimization step. The relative error of the yield estimation at the optimized design point x^* is computed by evaluating its relative deviation to the yield provided by 70000 MC simulations in HSPICE at the same design point and given as $Rel\ Err = \frac{|Yield(70000-sim)-Yield(2000-sim)|}{|Yield(70000-sim)|} \times 100$.

Table 4.3: Experimental results for the folded cascode amplifier

σ	Yield Eval (#)			Yield (%)		Rel Err (%)
	PGOpt	LOpt	Total	PGOpt	LOpt	
0.2	176	47	223	83.45	85.12	0.21
0.1	220	21	241	91.23	98.44	0.21
0.005	442	9	451	93.64	98.44	0.20

Using $\sigma = 0.1$, the proposed method locates the best yield solution. In this case, PGOpt reaches a near optimal solution with 220 yield evaluations. The local optimization needs to perform only 21 yield evaluations to converge to a higher quality design point. A close solution is reached with $\sigma = 0.005$. However, it requires 2X more yield evaluations. In fact, the value $\sigma = 0.1$ (i.e., 10% of the original search space size) offers a good trade-off between the solution optimality and the required number of yield estimations.

The proposed method finds a lower yield percentage with $\sigma = 0.2$. In this case, the sampling and subdivision strategy did not accurately locate the basin of convergence. Consequently, LOpt fails to locate a high yield solution. In fact, the result of the local refinement requires a good starting point. However, in all experiments, it uses a small number of yield evaluation. Its low computational cost is achieved thanks to the optimization of a non-expensive and local model of the yield and the simulation data reuse strategy.

Table 4.4: Experimental results of PGOpt applied solely on the folded cascode amplifier

σ	Yield Eval (#)	Yield (%)	Rel Err (%)
0.0001	656	98.43	0.20
0.0003	552	94.02	0.21

We apply PGOpt solely to locate the most robust design point with the optimum yield. The results are reported in Table 4.4. PGOpt applied with a low σ value succeeds in locating a good solution. However, it requires almost 3X higher number of yield evaluations, when compared to our approach with $\sigma = 0.1$. This observation confirms the slow convergence of the modified Lipschitiz optimization, despite its good search ability. The integration of a local refinement phase significantly decreases the number of yield evaluations and accelerates the optimization.

We compare our experimental results with high-ability algorithms including Genetic Algorithm (GA), Differential Evolution (DE) algorithm and GA-SA (Genetic Algorithm-Simulated Annealing), employed to optimize the yield for the cascode amplifier circuit. GA-SA uses GA as the global exploration mechanism and the simulated annealing (SA) algorithm to perform a local refinement. For all three methods, the feasible design space D_0 (i.e., the search space) is the same as the one used in the proposed yield optimization method. The evaluation of the yield is accomplished using MC simulations in HSPICE. For both GA and DE, the population size is 80 and the crossover rate is 0.8 [6]. The population is initialized by randomly selecting values of the design variables within D_0.

We executed 20 runs of each algorithm starting from 20 different initializations. Table 4.5 shows the best results in terms of yield quality among the 20 runs. We also include the result of the proposed method with $\sigma = 0.1$.

Table 4.5: Comparison with simulation-based stochastic search methods for the folded cascode amplifier

Method	Yield Eval (#)	Yield (%)	Rel Err (%)	Time [h]
Our method	241	98.44	0.19	2.65
GA	295	65.61	0.20	3.78
DE	275	70.98	0.20	3.75
GA-SA	319	83.11	0.19	3.83

The proposed optimization strategy is able to locate a higher yield rate with less computational time. The reduced computational time comes from: (1) the reduction of the search space allowed by the problem subdivision and the parallel computation; and (2) alleviating the slow convergence problem of the global search by the integration of a non expensive and linear local model-based optimization. Furthermore, the search ability of our approach obviously outperforms the stochastic optimization-based method thanks to an exhaustive exploration of potentially optimal regions. It can also be observed that neither DE nor the hybrid approach GA-SA is able to perform a reliable optimization, even though multiple runs were tried and the best optimization result is presented.

4.3.2 Two-stage Operational Amplifier

We consider a two-stage amplifier (op-amp) as shown in Figure 4.5. The length of all transistors is set to 130 nm. The number of independent design variables is 7 after applying the symmetry relations. The circuit specifications are shown in Table 4.6. The optimization search space is reported in Table 4.7. Any design point in the search space is guaranteed to satisfy the specification in nominal condition.

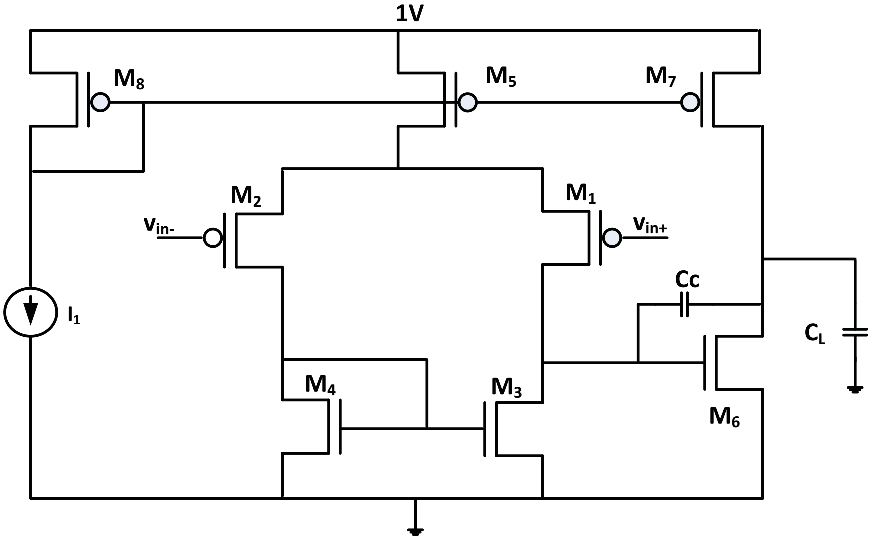

Figure 4.5: A Two-stage operational amplifier

Table 4.6: Set of specifications for the two-stage op-amp circuit

Perf metrics	Specification
$Av(dB)$	≥ 20
$GBW(MHz)$	≥ 3
$P_{DC}(mW)$	≤ 0.3
$PM(°)$	≥ 60
$DCOffset(mV)$	≤ 50

The local mismatch in each transistor pair is considered. It leads to a total of 12 process parameters. As shown in Table 4.8, the value $\sigma = 0.1$ is also offering the best trade-off in terms of solution quality and number of yield evaluation in this experiment. We notice again the dependence of the local optimization on the starting point. However, it always uses a very small number of yield evaluations. Also, our method is able to reach the same solution located by PGOpt applied with $\sigma = 0.0001$ (Table 4.9), but with 2X less yield evaluations. The relative error of the yield estimation at the optimized design point x^* is computed by evaluating its relative deviation to the yield provided by 70000 MC simulations in HSPICE at the same design point and given as $Rel\ Err = \frac{|Yield(70000-sim)-Yield(2000-sim)|}{|Yield(70000-sim)|} \times 100$.

Table 4.7: Design variables ranges of the two-stage op-amp circuit

Design variables	Ranges
$w_1 = w_2(\mu m)$	[1.5, 3.95]
$w_3 = w_4(\mu m)$	[1.02, 1.51]
$w_5 = w_8(\mu m)$	[3.1, 4.2]
$w_6(\mu m)$	[1.1, 2.3]
$w_7(\mu m)$	[2.13, 3.7]
$I_1(\mu A)$	[70, 80]
$C_c(pF)$	[7.5, 8.1]

Table 4.8: Experimental results for the two-stage op-amp circuit

σ	Yield Eval (#)			Yield (%)		Rel Err (%)
	PGOpt	LOpt	Total	PGOpt	LOpt	
0.2	178	5	183	78.05	80.02	0.19
0.1	219	8	227	94.03	98.34	0.18
0.005	395	9	404	95.64	98.34	0.18

We compare our experimental results with high-ability stochastic algorithms, including the Genetic Algorithm (GA), the Differential Evolution (DE) algorithm, the GA-SA and particle swarm optimization (PSO), employed to optimize the yield of the two stage op-amp circuit. For all four methods, we execute 30 runs of each algorithm starting from 30 different initializations. We include the experimental results of the highest yield quality results. The yield estimation is conducted in HSPICE and the results are summarized in Table 4.10. We also include the result of the current method with $\sigma = 0.1$.

Table 4.9: Experimental results of PGOpt applied solely on the two-stage op-amp

σ	Yield Eval (#)	Yield (%)	Rel Err (%)
0.0001	527	98.34	0.17
0.0003	486	96.90	0.18

The proposed method does not significantly reduce the number of yield evaluation compared with the stochastic search methods. However, it locates $\sim 10\%$ better

Table 4.10: Comparison with stochastic search methods for the two-stage op-amp

Method	Yield Eval (#)	Yield (%)	Rel Err (%)	Time [h]
Our method	227	98.34	0.18	0.91
GA	230	76.84	0.17	2.83
DE	211	85.27	0.16	2.76
GA-SA	252	87.98	0.19	2.91
PSO	103	61.98	0.18	2.37

quality of optimized yield rate and achieves ∼3X speedup when compared to the GA-SA. Indeed, GA-SA is the most efficient technique in terms of solution quality among the various search methods. We also noticed that the PSO method has a fast convergence ability but its search ability is very weak. Meanwhile, the proposed method is able to guarantee an acceptable level of error.

4.3.3 Three-stage Operational Amplifier

In this section, we apply the framework proposed in this thesis for variation-aware circuit sizing on a three-stage operational amplifier, shown in Figure 4.6 [64]. We also demonstrate that the method is capable of solving sizing problems with multiple conflicting performances specifications as well as high dimensional parameters space. Table 4.11 provides the specifications for the gain Av, gainbandwidth GBW, power P_{DC}, slew rate SR and DC offset voltage *DCOffset*.

Table 4.11: Set of specifications for the three-stage op-amp circuit

Perf metrics	Specification
$Av(dB)$	≥ 20
$GBW(MHz)$	≥ 3
$P_{DC}(mW)$	≤ 0.6
$SR(V/\mu s)$	≥ 30
DCOffset(mV)	≤ 40

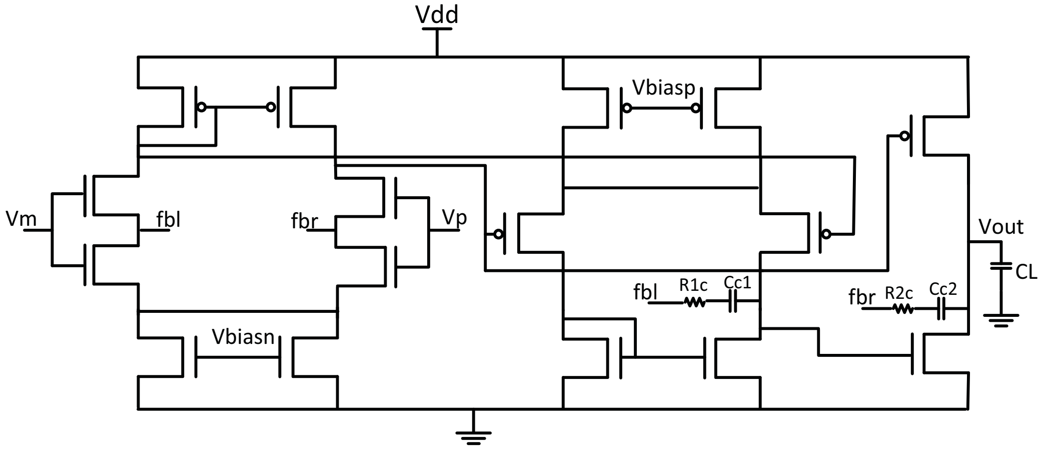

Figure 4.6: A Three-stage operational amplifier

The length of all transistors is fixed to 130nm. After applying the symmetry constraints, the number of independent design variables is 10. The nominal sizing procedure computes a continuous set of validated feasible design solutions that corresponds to the optimization search space D_0. The proposed two-phase optimization engine is applied to select the sizing solution with the highest yield rate. The global optimization phase stops the search when the size of the region with the best objective function at its center c_j reaches 10% of the original search space size. The local mismatch variables of each transistor pair are considered, which results in 56 process parameters. At each optimization iteration, surrogate models of the circuit performances are extracted and employed to estimate the yield rate.

We compare our experimental results with stochastic search optimization methods applied to size and optimize the yield of the three stage op-amp. The results are reported in Table 4.12. For our method, the column "Time Cost" is the run-time for all components in the proposed framework, including the nominal circuit sizing, the yield estimation and optimization. For all three stochastic search algorithms, we execute 10 runs starting from 10 different initializations. The crossover rate was manually improved through six runs. At each new run, the crossover rate was updated, trying to increase the yield rate when compared to the previous run. We include the results of the highest yield quality results. We also consider the design space defined by the technology library.

Table 4.12: Comparison with stochastic search methods for the three-stage op-amp circuit

Method	Yield Eval (#)	Yield (%)	Rel Err (%)	Time Cost [h]
Our method	291	98.97	0.23	5.35
GA	885	75.61	0.19	11.34
DE	825	78.98	0.19	11.25
GA-SA	975	87.88	0.19	11.49

According to Table 4.12, the yield result of the proposed method exhibits a small violation when compared to the yield estimation of 70000 MC runs in HSPICE as the performance models used in the yield evaluation do not totally match the circuit simulator-based performances evaluations. Still, the violation is small owing to the accuracy of the extracted sparse models.

Our method is able to locate higher quality of yield rate with less computational time. In fact, the nominal circuit sizing step decreases the design search space defined by the technology library. The restriction of the yield optimization to the space of feasible solutions avoids unnecessary yield evaluations and reduces the computational time. The model-based estimation of the yield rate also reduces the run-time at the cost of a slight increase in the relative error.

Despite being recognized as an effective evolutionary search engine for global optimization, the DE algorithm reaches a sub-optimal solution. On the other hand, the GA-SA method uses the SA method to perform a local optimization starting from the best design point located by the GA. However, the yield of the located design point is 10% less than the optimized yield reached by our method.

4.4 Summary

The aim of yield optimization is to find the design point that has the maximum yield, considering the manufacturing variation. The search space which is determined by the technological process is very large. Conducting the yield optimization on the complete search space would be inefficient and time consuming, as many design points cannot satisfy the specifications even for nominal values of process parameters. In this chapter, we have employed the SMT-based circuit sizing methodology presented in Chapter 2 to characterize the feasible design solutions. Then, we proposed a novel method for analog yield optimization that aim at selecting the most robust design point. The technique samples the most potential region of the feasible design space and locates a design point near the optimal solution. A local model-based local search is then integrated to highly speedup the convergence. Its efficiency is elevated by the reuse of existing simulation data of the global search phase. Compared with simulation-based stochastic optimization, our method identifies more robust design points (i.e., with higher yield rate) within less run-time and without largely affecting the accuracy. Furthermore, it does not require multiple runs and less parameters need to be set.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Analog circuit sizing consists in determining the device sizes and biasing voltages and currents such that the circuit meets its specifications. Yield analysis estimates the probability that the circuit meets its specification under parameters fluctuation. Available methodologies for variation-aware analog circuit sizing are based on the integration of a performance and yield estimator with an optimization technique. Despite the huge progress made in analog design automation and research area, circuit sizing is not trivial and many challenges still exist. For example, available optimization techniques cannot guarantee an exhaustive coverage of the design search space and hence, are not able to ensure high quality design solutions. Furthermore, existing yield analysis methods can be computationally expensive.

In this thesis, we proposed a framework for analog circuits sizing in nominal condition, yield estimation and yield optimization. We proposed several new techniques and algorithms to tackle specific limitations of existing methods. The first contribution of this thesis is the development of a nominal sizing procedure that ensures an exhaustive coverage of the design space and outputs guaranteed bounds on the feasible design solutions. To do so, we characterized transistor small signal parameters as a function of transistor biases (voltage and current) using simulation data and polynomial regression. Given the circuit topology and the specification properties, the sizing problem is encoded using nonlinear constraints. We employed an SMT solver and interval arithmetic techniques to track a conservative approximation of the circuit operating point and to determine all possible reachable performances. A search space subdivision approach and a parallel exploration efficiently accelerate our proposed solution scheme. The SMT-based approach ensures a complete coverage of the design space and is able to locate higher quality solutions when compared to existing methods. A continuous range of each device dimension is determined out of the set of operating point using efficient modeling and global optimization approaches. Our method is able to characterize continuous sets of transistor sizing variables for which the circuit meets the target specifications with high confidence.

The second contribution is the development of a new method for fast and efficient computation of parametric yield that combines the advantages of sparse regression and SMT solving techniques. The method constructs sparse polynomial surrogate models based on LASSO to find a low degree polynomial approximations of the circuit performances. A procedure inspired by statistical model checking is then introduced to verify the model accuracy. The resulting model can be viewed as a statistically guaranteed model of the circuit behavior. An SMT-based solving technique is then employed to find all likely failure regions in the parameters space. The yield calculation is based upon a geometric calculation of probabilistic hypervolumes subtended by the fail regions in the parameters space. The proposed method is computationally efficient and is suitable for handling problems with tens of process parameters.

The third contribution of this thesis is the elaboration of a new search strategy for yield optimization. First, a parallelized global optimization phase uses the modified Lipshitizian optimization method to locate the basin of convergence of the optimum solution. The search ensures that potentially optimal regions of the feasible design space are not omitted. The guarantee is obtained by ensuring that the largest unexplored regions in the search space are small enough. Once a good approximation of the global optimum is located, it is exploited by the local optimization phase. The local search is integrated to remedy to the limitation of the Lipshitizian method by accelerating its convergence speed. It builds interpolating models using a linear combinations of Radial Basis Functions that approximates locally the objective function and conducts a local refinement. Its efficiency is further elevated by the reuse of existing simulation data of the global search phase.

We applied the developed methods for the analysis of various analog circuits and compared the results of the proposed methods to the existing approaches. The application of the SMT-based nominal circuit sizing method on a two-stage amplifier and a folded cascode amplifier shows its high ability to guarantee an exhaustive coverage of the search space design and to meet the performance constraints, when compared with high-ability optimization algorithms. The proposed method provides the first steps towards the integration of formal techniques for analog synthesis. The experimental results are very promising. However, they can be further enhanced by automating the extraction of the performance constraints and extending it to handle larger circuits. Furthermore, compared with existing methods, the application of the surrogate-based yield estimation method on various analog circuits shows that it is able to handle yield problems with many process parameters and extremely high yield rate. However, efficient heuristics and parallelization techniques should be considered because the computational cost of the surrogate modeling algorithm may increase if the number of process parameters and the number of performance metrics largely increase. Besides, the application of the yield optimization strategy shows that the method is able to minimize the risk of missing potentially optimal design points and does not require multiple runs when compared to stochastic optimization techniques. The optimization strategy can be further enhanced by making it tractable for larger circuits and including more aggressive process variation. Some of the mentioned future enhancements and directions of further research are detailed in the next section.

5.2 Future Work

Based on the work presented in this thesis, several enhancements and directions of further research can be pursued.

In our nominal circuit sizing methodology, we employed an operating point driven circuit sizing technique. That is, the circuit operating point is first selected for a fixed transistor length, then converted to transistors widths. Indeed, the circuit performances are modeled as a function of transistor biases (voltage and current) and for fixed transistors length (e.g., minimum length, twice the minimum length, etc.). However, fixing the transistors length in our sizing constraints may have a large effect on the performances of the circuit. This limitation may be solved by using advanced modeling techniques for large scale problems that allow the inclusion of the length as a design variable. In fact, the length parameter has a large range that is allowed by the technology. Using conventional techniques for regression alike least square polynomial regression may easily lead to over fitting. It is possible to combine sparse regression with powerful machine learning techniques such as random forests and bootstrap aggregating to learn piecewise approximations and ovoid over fitting. In this case, the circuit performances can be modeled as a function of the device biases and length. Consequently, the length is considered as a design variable and higher performances can be reached.

Accurate and not complex surrogate models can replace transistor-level simulation and significantly fasten the performances assessment and consequently the yield estimation. However, the computational cost for polynomial regression-based performance modeling increases with: (1) the number of process parameters (i.e., the number of variables); and (2) the order of the trained polynomial model. In the case where the dimensionality is extremely high, the proposed adaptive regression technique must choose a set of important polynomial terms from numerous (e.g., millions of) possible candidates; and hence, the surrogate model training algorithm described in this thesis may become computationally unaffordable. We believe that this limitation can be addressed by the integration of efficient heuristics and parallel computing. For example, MATLAB offers parallel computing features that can solve computationally and data-intensive problems using multicore processors, computer clusters and parallel for loops.

The proposed method for nominal circuit sizing is efficient in the sense that it ensures an exhaustive coverage of the design search space. However, it is designed to handle circuits with a small to medium number of transistors. It is possible to handle larger circuits by developing a hierarchical sizing technique. First, the analog circuit is decomposed into a set of smaller sub-circuits which decreases the number of variables involved in the SMT problems. In this case, we need to handle the correlations between the partitioned sub-circuits. Second, each SMT problem associated with a sub-circuit can be further decomposed into a set of sub-problems with less constraints to further improve the efficiency.

For the yield optimization, we have to apply the optimization strategy in the presence of more severe process variations. Global process variation can be considered with local mismatch. The IC foundry TSMC offers variational device libraries that model global variation. It would be interesting to verify the circuits under both types of variations. However, the computational cost of the yield optimization will largely increase. Also, for instance, the global optimization problem is divided into subproblems that are solved independently and in parallel. It is possible to set a global communication strategy between the different subproblems in order to minimize the computational cost and to avoid unnecessary optimizations iterations. Moreover, the stopping criteria of the global optimization phase is set manually by the user. The criteria should balance between the computational cost and the nearness of the solution to the optimal one. This part can be enhanced by dynamically varying the stopping criteria and by adding a constraint related to the computational cost and defined in terms of the number of yield evaluations.

Bibliography

[1] D. Boolchandani, A. Kumar, and V. Sahula. Multi-objective genetic approach for analog circuit sizing using svm macro-model. *IEEE Region 10 Conference*, pages 1–6, 2009.

[2] B. Liu, F.V. Fernández, G. Gielen, R. Castro-López, and E. Roca. A memetic approach to the automatic design of high-performance analog integrated circuits. *ACM Transactions on Design Automation of Electronic Systems*, 14(3):1–42, 2009.

[3] R. R. Schaller. Moore's law: past, present and future. *IEEE Spectrum*, 34(6):52–59, Jun 1997.

[4] M. Bhushan and M. B. Ketchen. *CMOS Test and Evaluation*. Springer, 2015.

[5] P. G. Van der Plas, G. Gielen, and W. Sansen. *A Computer-Aided Design and Synthesis Environment for Analog Integrated Circuits*. Springer, 2006.

[6] B. Liu, G. Gielen, and F.V Fernández. *Automated Design of Analog and High-frequency Circuits*. Springer, 2014.

[7] SPICE. Spice, user's guide and manuals, December 2014.

[8] H. E. Graeb. *Analog Design Centering and Sizing*. Springer, 2007.

[9] M. Wirnshofer. *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits*. Springer, 2013.

[10] O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar. Enabling the DC solutions characterization using a fuzzy approach. *New Circuits and Systems Conference*, pages 161–164, 2014.

[11] B. Liu, F. V. Fernández, and G. G. E. Gielen. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(6):793–805, 2011.

[12] T. Mukherjee, L. R. Carley, and R. A. Rutenbar. Efficient handling of operating range and manufacturing line variations in analog cell synthesis. *IEEE Transactions on Computeer-Aided Design of Integrated Circuits and Systems*, 19(8):825–839, 2000.

[13] M. Fronzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.

[14] iSAT3. Tight integration of satisfiability and constraint solving. `http://isat.gforge.avacs.org/`, 2017.

[15] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury. Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs. *Proceedings of the IEEE*, 95(3):640–669, 2007.

[16] D. Han and A. Chatterjee. Simulation-in-the-loop analog circuit sizing method using adaptive model-based simulated annealing. *International Workshop on*

System-on-Chip for Real-Time Applications, pages 127–130, 2004.

[17] B. Peng, F. Yang, C. Yan, X. Zeng, and D. Zhou. Efficient multiple starting point optimization for automated analog circuit optimization via recycling simulation data. pages 1417–1422, March 2016.

[18] A. Bagirov, N. Karmitsa, and M. M. Mkel. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer, 2014.

[19] L. Xin, P. Gopalakrishnan, X. Yang, and L. T. Pileggi. Robust analog/RF circuit design with projection-based posynomial modeling. *International Conference on Computer Aided Design*, pages 855–862, 2004.

[20] Y. Wang, M. Orshansky, and C. Caramanis. Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization. *Design Automation Conference*, pages 164:1–164:6, 2014.

[21] G. Huang, L. Qian, S. Saibua, D. Zhou, and X. Zeng. An efficient optimization based method to evaluate the DRV of SRAM cells. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(6):1511–1520, 2013.

[22] P. Cheng, H. Ming, and C. Chih. Page: Parallel agile genetic exploration towards utmost performance for analog circuit design. *Design, Automation Test in Europe*, pages 1849–1854, 2013.

[23] J. Suykens and J. Vandewalle. *Least Squares Support Vector Machine Classifiers*. Kluwer Academic Publishers, 1999.

[24] A. Lemke, L. Hedrich, and E. Barke. Analog circuit sizing based on formal methods using affine arithmetic. *International Conference on Computer Aided Design*, pages 486–489, 2002.

[25] I. G. Gomez, T. McConaghy, and E. T. Cuautle. Operating-point driven formulation for analog computer-aided design. *Analog Integrated Circuits and Signal Processing*, 74(2):345–353, 2013.

[26] M. K. Hsuan, P. P. Cheng, and C. H. Ming. Integrated hierarchical synthesis of analog/RF circuits with accurate performance mapping. *International Symposium on Quality Electronic Design*, pages 1–8, 2011.

[27] P. Mandal and V. Visvanathan. CMOS op-amp sizing using a geometric programming formulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(1):22–38, 2001.

[28] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration, the VLSI Journal*, 42(2):137–148, 2009.

[29] M. H. Zaki, I. M. Mitchell, and M. R. Greenstreet. DC operating point analysis: A formal approach. *Workshop on Formal Verification of Analog Circuits*, 2009.

[30] F. Leyn, G. Gielen, and W. Sansen. An efficient DC root solving algorithm with guaranteed convergence for analog integrated cmos circuits. *International Conference on Computer-Aided Design*, pages 304–307, 1998.

[31] B. Liu, M. Pak, X. Zheng, and G. Gielen. A novel operating-point driven method for the sizing of analog IC. *International Symposium on Circuits and Systems*, pages 781,784, 2011.

[32] V. Boos, J. Nowak, M. Sylvester, S. Henker, S. Hoppner, H. Grimm, D. Krausse, and R. Sommer. Strategies for initial sizing and operating point analysis of analog circuits. *Design, Automation and Test in Europe*, pages 1–3, 2011.

[33] C. Jacoboni and P. Lugli. *The Monte Carlo Method for Semiconductor Device Simulation*. Springer Vienna, 1998.

[34] S. Sun, Y. Feng, C. Dong, and X. Li. Efficient SRAM failure rate prediction via gibbs sampling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(12):1831–1844, 2012.

[35] J. Jaffari and M. Anis. On efficient LHS-based yield analysis of analog circuits. *Computer-Aided Design of Integrated Circuits and Systems*, 30(1):159–163, 2011.

[36] J. Yao, Z. Ye, and Y. Wang. Importance boundary sampling for sram yield analysis with multiple failure regions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(3):384–396, 2014.

[37] A. Singhee and R. A. Rutenbar. Why quasi-monte carlo is better than monte carlo or latin hypercube sampling for statistical circuit analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1763–1776, 2010.

[38] S. Sun, X. Li, H. Liu, K. Luo, and B. Gu. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(7):1096–1109, 2015.

[39] F. Gong, H. Yu, Y. Shi, D. Kim, J. Ren, and L. He. Quickyield: An efficient global-search based parametric yield estimation with performance constraints. *Design Automation Conference*, pages 392–397, 2010.

[40] B. Liu, F. V. Fernández, and G. E. Gielen. Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational

intelligence techniques. *IEEE Transactions on CAD*, 30(6):793–805, 2011.

[41] B. Liu, Q. Zhang, F. V. Fernández, and G. E. Gielen. An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization. *IEEE Transactions on Evolutionary Computation*, 17(6):786–796, 2013.

[42] R. Schwencker, F. Schenkel, M. Pronath, and H. Graeb. Analog circuit sizing using adaptive worst-case parameter sets. *Design, Automation and Test in Europe*, pages 581–585, 2002.

[43] MATLAB. Documentation center. `http://www.mathworks.com/products/matlab/`, 2017.

[44] M. Keramat and R. Kielbasa. Modified latin hypercube sampling Monte Carlo (MLHSMC) estimation for average quality index. *Analog Integrated Circuits and Signal Processing*, 19(1):87–98, 1999.

[45] S. M. Rump. *Developments in Reliable Computing*. Springer, 1999.

[46] S. K. Tiwary, A. Gupta, J. R. Phillips, C. Pinello, and R. Zlatanovici. First steps towards SAT based formal analog verification. *International Conference on Computer-Aided Design*, pages 1–8, 2009.

[47] N. Dong and J. Roychowdhury. Piecewise polynomial nonlinear model reduction. *Design Automation Conference*, pages 484–489, 2003.

[48] J.A. Momoh and J.Z. Zhu. Improved interior point method for OPF problems. *IEEE Transactions on Power Systems*, 14(3):1114–1120, 1999.

[49] R. J. Baker. *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 2010.

[50] W. Liu, X. Jin, J. Chen, M-C. Jeng, Z. Liu, Y. Cheng, K. Chen, M. Chan, K. Hui, J. Huang, R. Tu, P.K. Ko, and C. Hu. BSIM 3v3.2 MOSFET model users' manual. Technical report, EECS Department, University of California, Berkeley, USA, 1998.

[51] C. Zhang and Y. Ma. *Ensemble Machine Learning*. Springer, 2012.

[52] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[53] C. Gu and J. Roychowdhury. *Yield Estimation by Computing Probabilistic Hypervolumes. In: Extreme Statistics in Nanoscale Memory Design*, pages 137–177. Springer, 2010.

[54] M. Robnik-Sikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression. *International Conference on Machine Learning*, pages 296–304, 1997.

[55] K. Kira and L. A. Rendell. A Practical Approach to Feature Selection. *International Conference on Machine Learning*, pages 249–256, 1992.

[56] X. Li. Finding deterministic solution from underdetermined equation: Large-scale performance variability modeling of analog/RF circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1661–1668, 2010.

[57] H. Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[58] H. L. S. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Computer Science Department, Carnegie Mellon

University, Pittsburgh, Pennsylvania, USA, 2005.

[59] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 06 1945.

[60] TSMC 65*nm* CMOS Process Technology. `http://www.tsmc.com`, 2017.

[61] P.R. Bevington and D.K. Robinson. *Data reduction and error analysis for the physical sciences*. McGraw-Hill, 2003.

[62] E. Seevinck, F. J. List, and J. Lohstroh. Static-noise margin analysis of mos sram cells. *IEEE Journal of Solid-State Circuits*, 22(5):748–754, 1987.

[63] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan. Breaking the simulation barrier: Sram evaluation through norm minimization. *International Conference on Computer-Aided Design*, pages 322–329, 2008.

[64] V. Saxena and R. J. Baker. Indirect compensation techniques for three-stage cmos op-amps. *Midwest Symposium on Circuits and Systems*, pages 9–12, 2009.

[65] R. Horst and P.M. Pardalos. *Handbook of global optimization*. Kluwer Academic Publishers, 1995.

[66] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[67] S. M. Wild, R. G. Regis, and C. A. Shoemaker. Orbit: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

Biography

Education

- **Concordia University**: Montreal, Quebec, Canada

  Ph.D. degree, Dept. of Electrical & Computer Engineering, (September 2012 - April 2017)

- **National Engineering School of Tunis**: Tunis, Tunisia

  Master's degree in Communication Systems, (September 2010 - June 2011)

- **National Engineering School of Tunis**: Tunis, Tunisia

  Diploma degree in Telecommunication Engineering, (September 2006 - June 2011)

Awards

- Concordia University Accelerator Award, Concordia University, Canada (2017).
- Concordia University Conference and Exposition Award, Concordia University, Canada (2017).

- Best Paper Award in Hardware Verification Group, Concordia University, Canada (2016).
- Best Paper Award in Hardware Verification Group, Concordia University, Canada (2015).
- Concordia University Conference and Exposition Award, Concordia University, Canada (2015).
- Design Automation Conference Certificate of Appreciation, California, USA (2015).
- Tuition Fee Waiver for Ph.D Program, Canada (2012-2015).
- Ranked first in the Diploma Degree of Telecomunication, National Engineering School of Tunis, Tunisia (2011).

Work History

- **Teaching Assistant**, Dept. of Electrical & Computer Engineering, Concordia University, Montreal, Quebec, Canada (2016-2017)
- **Research Assistant**, Dept. of Electrical & Computer Engineering, Concordia University, Montreal, Quebec, Canada (2011-2017)

Publications

- **Journal Papers**
  - **Bio-Jr1** O. Lahiouel, M. H. Zaki, and S. Tahar, Accelerated and Reliable Analog Circuits Yield Analysis using SMT Solving Techniques; IEEE Transactions on CAD of Integrated Circuits and Systems, DOI:10.1109/TCAD.2017.2651807, January 2017, pp. 1-14.
  - **Bio-Jr2** O. Lahiouel, M. H. Zaki, and S. Tahar, Exploiting Bounds Optimization for the Semi-formal Verification of Analog Circuits; Integration, the VLSI Journal. (Accepted with minor modification), November 2016, pp. 1-14.
- **Refereed Conference Papers**
  - **Bio-Cf1** O. Lahiouel, M. H. Zaki, and S. Tahar, Enhancing Analog Yield Optimization for Variation-aware Circuits Sizing. [Proc. Design Automation and Test in Europe (DATE'17), Lausanne, Switzerland, March 2017, pp. 1-4.]
  - **Bio-Cf2** O. Lahiouel, M. H. Zaki, and S. Tahar, Towards Enhancing Analog Circuits Sizing Using SMT-based Techniques. [Proc. Design Automation Conference (DAC '15), San Francisco, California, USA, June 2015, pp. 1-6.]
  - **Bio-Cf3** O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar, Enabling the DC Solutions Characterization using a Fuzzy Approach. [Proc. IEEE New Circuits and Systems Conference (NEWCAS'14), Trois-Rivieres, Quebec, Canada, June 2014, pp. 161-164.]

- **Bio-Cf4** O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar, A Semi-Formal Approach for Analog Circuits Behavioral Properties Verification. [Proc. Great Lakes Symposium on VLSI (GLSVLSI'14), Houston, Texas, USA, May 2014, pp. 247-248.]

- **Refereed Workshop Presentations**
  - **Bio-Ws1** O. Lahiouel, M. H. Zaki, and S. Tahar, A Framework for Variation-Aware Analog Circuits Sizing, University Booth, Design Automation and Test in Europe (DATE'17), Lausanne, Switzerland, March 2017.
  - **Bio-Ws2** O. Lahiouel, M. H. Zaki, and S. Tahar, Enhancing Yield-aware Analog Circuits Sizing. ReSMIQ-Meiji-ISEP Workshop, International Symposium on Circuits and System (ISCAS '16), Montreal, Quebec, Canada, May 2016.
  - **Bio-Ws3** O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar, A Tool for modeling and analysis of analog circuits, University Booth, Design Automation and Test in Europe (DATE'12), University Booth, Dresden, Germany, March 2012.